

NAVAL POSTGRADUATE SCHOOL

Monterey, California



**Teaching Introductory Computer Security
at a
Department of Defense University**

by

Cynthia E. Irvine
Roger Stemp
Daniel F. Warren

April 1997

Approved for public release; distribution is unlimited.

Prepared for:

Naval Postgraduate School
Monterey, California 93943

19970710 023

DTIC QUALITY INSPECTED 4

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M. J. Evans
Superintendent

Richard Elster
Provost

This report was prepared as part of the Naval Postgraduate School Center For Information Systems Security (INFOSEC) Studies and Research (NPS CISR) at the Naval Postgraduate School, which is currently funded by the National Security Agency under Contract No. H98230-R297-0030. Any opinions, findings, and conclusions or recommendations expressed in this report are those of the authors and do not necessarily reflect the views of the National Security Agency.

Cynthia E. Irvine
CYNTHIA E. IRVINE
Assistant Professor
Department of Computer Science

Reviewed by:

Neil C. Rowe

NEIL C. ROWE
Associate Professor
Department of Computer Science

Ted Lewis

TED LEWIS
Chairman
Department of Computer Science

Released by:

D.W. Netzer

NETZER
Dean of Research

REPORT DOCUMENTATION PAGE

Form approved

OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1315 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
		2 APRIL 1997	PROGRESS- FROM 1/1/97 TO 3/31/97	
4. TITLE AND SUBTITLE Teaching Introductory Computer Security at a Department of Defense University			5. FUNDING H98230-R297-0030	
6. AUTHOR(S) Cynthia E. Irvine, Roger Stemp, and Daniel F. Warren				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Computer Science Department Naval Postgraduate School Monterey, CA 93943			8. PERFORMING ORGANIZATION REPORT NUMBER NPSCS-97-002	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Security Agency 9800 Savage Road Fort George G. Meade, MD 20755			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Naval Postgraduate School Center for Information Systems Security (INFOSEC) Studies and Research (NPS CISR) has developed an instructional program in computer security. Its objective is to insure that students not only understand practical aspects of computer security associated with current technology, but also learn the fundamental principles that can be applied to the development of systems for which high confidence in policy enforcement can be achieved. Introduction to Computer Security, the cornerstone course for our program is described here.				
14. SUBJECT TERMS computer security, INFOSEC, education			15. NUMBER OF PAGES 386	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED	

NSN 7540-01-280-5800

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18

Enclosure (6)

Teaching Introductory Computer Security at a Department of Defense University

Cynthia E. Irvine, Roger Stemp, and Daniel F. Warren

Naval Postgraduate School
Department of Computer Science
Monterey, California 93943-5118

April 1997

Abstract

The Naval Postgraduate School Center for Information Systems Security (INFOSEC) Studies and Research (NPS CISR) has developed an instructional program in computer security. Its objective is to insure that students not only understand practical aspects of computer security associated with current technology, but also learn the fundamental principles that can be applied to the development of systems for which high confidence in policy enforcement can be achieved. Introduction to Computer Security, the cornerstone course for our program, is described here.

1 Introduction

Twenty-five years ago, computers were still largely monolithic mainframes, physically isolated from cyber-predators and closely tended by dedicated staffs of technical and administrative personnel. Even then, when computers were the domain of scientists and engineers, the need for computer security was recognized [22] and programs to achieve it pursued, e.g. [23].

Now society's relationship with the computer has changed dramatically. Computers are everywhere: in our tools and appliances, in our homes, schools and offices. They are used to manage our money and all phases of commercial enterprise. The evolution of techniques with which to download executable content either for work or entertainment from remote systems provides powerful mechanisms that tie together the far corners of the world as never before. Now computer security is no longer an esoteric subject discussed by a small group of academics and system administrators, but a topic that must be appreciated by all citizens of the information age. The education of computer security professionals is critical to the support of personal, corporate, and government information systems security objectives.

Over the past six years, at the Naval Postgraduate School (NPS), we have developed a program in INFOSEC education. This effort is under the umbrella of the Naval Postgraduate School Center for Information Systems Security Studies and Research and Research (NPS CISR). A cornerstone of the educational program offered by NPS CISR is the introductory course in computer security that we have developed. This report is intended to provide background regarding the rationale for the course's content and a detailed description of the course itself.

2 Background

2.1 Computer Science at NPS

The INFOSEC education program at NPS is part of the Computer Science Curriculum. In the two-year, eight-quarter Masters degree program, students are required to demonstrate competence in a core curriculum of traditional computer science courses. Many entering students have no prior education in computer science. They must cover the fundamentals of computer science which include the theory of formal languages, computer systems principles, object-oriented programming, data structures, artificial intelligence, operating systems, software methodology, database systems, computer communications and networks, computer graphics or interactive computation, computer security, and the design and analysis of algorithms.

To allow for specialization in a variety of areas, the core curriculum is enhanced with tracks in the following areas: software engineering; artificial intelligence and robotics; database and data engineering; computer graphics and visual simulation; computer systems and architecture; and computer security.

Each student's course of study is capped by a written thesis, most often based on research directed by a faculty member in the student's chosen specialization track. This work must be conducted during the sixth through eighth quarters in conjunction with classes. Thesis research allows students to be involved in work addressing an unsolved problem, usually within the framework of the U.S. Department of Defense (DoD) or U.S. Government; it enhances both their oral and written presentation skills, and it hones their critical thinking abilities. In many cases students start thesis research prior to the sixth quarter.

2.2 NPS CISR

The computer security track was established in 1991 to address the growing need for INFOSEC education of U.S. military officers. First, a two-course sequence in INFOSEC was offered: an introductory course and an advanced topics course. In 1994 the track was expanded and new INFOSEC courses were added to the Computer Science Curriculum.

With the encouragement of sponsors, the Naval Postgraduate School Center for INFOSEC Studies and Research was officially established in October 1996. Today, NPS CISR involves the research of eight faculty and staff members, nine thesis students, and approximately 150 students participating in classes and laboratory work annually. Students in Computer Science, Information Technology Management, and Information Warfare curricula all take courses in computer security.

NPS CISR serves the INFOSEC research and education needs of DoD/DoN in the following primary areas.

- Curriculum development ensures that a coherent and comprehensive program in INFOSEC foundations and technology is presented at the university and postgraduate levels.
- Development of the INFOSEC and Trusted Systems Laboratory supports the INFOSEC teaching and research programs at NPS.
- Faculty development fosters the insertion of INFOSEC concepts at appropriate points in general computer science courses and involves interested faculty members in leading-edge INFOSEC research problems.

- A Visiting Professor program which brings INFOSEC experts to NPS to offer courses and engage in research with faculty and students.
- An Invited Lecture series injects commercial and military relevance into the NPS CISR activities.
- An academic outreach program permits other non-CISR academic institutions to benefit from the INFOSEC education and research developments at NPS.
- An effort to insure that NPS CISR graduates are identified so that their expertise can be applied to the wide variety of INFOSEC challenges in DoD and U.S. Government.
- Research, focusing on INFOSEC problems, with emphasis on those of DoN, DoD, and U.S. Government.

2.3 INFOSEC Curriculum

The curriculum for the INFOSEC track has been designed to meet the following general objectives:

- To provide courses for both beginning and advanced students,
- To provide courses accessible by students who are not in the Computer Science curriculum,
- To insure that Computer Science students have a strong foundation upon which to base advanced course work in computer science and INFOSEC,
- To involve students in ongoing research and technology development efforts associated with computer security and INFOSEC, and
- To enhance students' laboratory experience through the hands-on use of secure systems,
- To heighten awareness of security issues with non-computer science majors, such as those studying management or procurement.

2.3.1 NPS CISR Curriculum Philosophy

To teach computer security, an accurate definition of the subject is needed. At the most general level, security pertains to access either to computational resources or to information in a computer system. Access to computational resources can be denied to legitimate users through the disruption of service, theft, or merely too little processing power or bandwidth for the amount of computation required. In contrast, information is vulnerable to unauthorized modification or disclosure. Access to information is controlled to prevent unauthorized modification and disclosure. Thus we have a triad of INFOSEC objectives:

- Availability: to ensure that information and/or resources are not being withheld in an unauthorized manner.
- Confidentiality: to ensure that information is not disclosed in an unauthorized manner.
- Integrity: to ensure that information is not modified in an unauthorized manner.

It is important to clearly separate problems in availability from those associated with confidentiality and integrity. For availability, we wish to ensure access to a resource, whereas, for

the other two, we wish to permit only authorized parties access to information. Students learn that availability is subjective, what is sufficient access to resources for one individual may be inadequate for another. Thus it is difficult to express an availability policy. In contrast confidentiality and integrity can be precisely defined and it is possible to know when a system has provided the necessary and sufficient mechanism to support either a confidentiality or an integrity policy, or both.

In terms of content, we believe that it is essential that students understand the fundamental concepts behind risk avoidance as articulated in the Reference Monitor Concept [4]. This encompasses a notion of completeness that is absent from more intuitive and/or ad hoc approaches to computer security. The idea that a policy enforcement mechanism is always invoked, cannot be modified by unauthorized individuals, and is inspectable so that one can assess whether or not it works correctly is applicable over a broad range of security policies and mechanisms. This requires systematic presentation of the principles of computer security and a corresponding engineering discipline. The feasibility of designing systems which are less susceptible to recurrent cycles of penetrations and patches [17] can be described and demonstrated.

In addition, our students must know how to function in the real world, where risk management techniques are employed [1]. The practical nature of these approaches make them attractive in situations where more complete systems are not in place. (Note that we are making a distinction between the study of these protection functions and system maintenance.) Issues associated with the incremental achievement of security objectives must be addressed.

Topics have been identified which we believe should be covered in an INFOSEC education program. Our position as a DoD university is reflected in some of these subjects, however, most are universal. They include, in no particular order: Risk Analysis, Disaster Recovery, Access Controls and Authentication, System Maintenance, Cryptography, Emanations Security, Audit Management, Protocols, Key Management, Configuration Management and Backups, Privacy Issues, User Monitoring, Personnel Issues, Physical Security. Additional topics are covered as needed. Coverage in the introductory survey courses, by necessity, must be broad rather than deep, but the survey must provide sufficient technical depth to serve as a springboard for progressing to advanced studies.

2.4 Lab Requirements

The ultimate objective of all INFOSEC studies is to improve security in real systems. Thus, practical laboratory experience is crucial for an effective INFOSEC program. Laboratory exercises in the form of tutorials and projects help to reinforce and extend concepts conveyed in lectures as well as help prepare students for effective thesis research.

Most NPS CISR courses include a lab component. As existing courses are refined and new ones developed, corresponding lab exercises are prepared or updated. An objective of the NPS CISR program is to allow students to understand the kinds of technologies that are available to solve current computer security problems and to consider potential future technologies. Students are given first-hand experience in using a variety of trusted systems and explore topics in security policy enforcement, security technology for database systems, monolithic and networked trusted computing techniques, and tools to support the development of trusted systems.

3 INFOSEC Curriculum

The INFOSEC courses for computer science students is integrated as a specialization track within the core computer science curriculum. The course matrix for the track is shown in Table 1.

Table 1: Computer Security Track of NPS Computer Science Curriculum.

1st Quarter (Fall or Spring)	CS-2970 (3-2) Object-Oriented Programming 1	CS-3010 (4-0) Computing Devices and Systems	MA-3025 (5-1) Logic and Discrete Mathematics	MA-3030 (5-1) Intro. to Combinatorics & Its Applications	
2nd Quarter (Winter or Summer)	CS-2972 (3-2) Object-Oriented Programming 2	CS-3300 (3-2) Data Structures	CS-3200 (3-2) Introduction to Computer Architecture	CS-3601 (4-0) Theory of Formal Languages & Automata	
3rd Quarter (Spring or Fall)	CS-3701 (3-2) Object-Oriented Programming in C++	CS-3650 (4-0) Theory of Algorithms	CS-3600 (3-2) Introduction to Computer Security	CS-3460 (3-1) Software Methodology	CS-4900 (2-0) Research Seminar in Computer Science
4th Quarter (Summer or Winter)	CS 3310 (4-0) Artificial Intelligence	CS 3320 (3-1) Database Systems	CS-3450 (3-2) Operating Systems	CS-3111 (4-0) Principles of Programming Languages	CS 4905
5th Quarter (Fall or Spring)	CS3502 (4-0) Computer and Communications Networks	CS-3651(4-0) Computability Theory and Complexity	CS-4600 (3-2) Secure Systems	CS-3670 (3-2) Management of Secure Systems	
6th Quarter (Winter or Summer)	CS 4203 (3-2) Interactive Computation Systems	Thesis	CS-4605 (3-1) Policies, Models and Formal Methods	CS-4112 (3-2) Distributed Operating Systems	
7th Quarter (Spring or Fall)	NS-3252 (4-0) Joint & Maritime Strategic Planning	Thesis	CS 4602 (4-0) Adv. Computer Security (Database Security)	Track Requirement	Note: International students replace NS-3252 with IT-1500.
8th Quarter (Summer or Winter)	Thesis	Thesis	CS-4614 (3-1) Advanced Topics in Computer Security	CS 3690 (4-0) App. Info. Sec. Systems (Network Security)	
<p>Bold Outline indicates courses specifically required for the Computer Security Track</p> <p>The track requirement in the seventh quarter is determined as appropriate based on the thesis research and interests of the individual student.</p>					

3.0.1 Introduction to Computer Security

Two courses, Introduction to Computer Security and Management of Secure Systems, provide an overview of INFOSEC principles and techniques described in section 2.3 . The two courses review both the conceptually complete and more intuitive approaches to INFOSEC. These provide the students with an appreciation of both foundational concepts and current practice in computer security.

Introduction to Computer Security was the first course offered at NPS. Over time, we have made significant changes to the NPS CISR flagship course, Introduction to Computer Security. When initially offered, it was an upper level graduate course and had daunting prerequisites: data structures, software system design, networks, databases, and software methodology. It included many of the topics now covered by the two current courses, Introduction to Computer Security and Management of Secure Systems. The original course skimmed many topics, but there was still insufficient time to survey all areas of computer security deemed important. Therefore, we decided to create two courses: one on the principles and underlying mechanisms for system security and the other on practical aspects of structuring and maintaining secure systems. In 1995, Introduction to Computer Security was modified to be an intermediate rather than an upper-level graduate course. Several benefits accrue from this change. With fewer prerequisites, the course is accessible by a much larger population of NPS students. This results in an increased number of DoD personnel having taken a graduate-level INFOSEC course. In addition, it may be taken much earlier in each students' course of study. Thus students are "sensitized" to INFOSEC issues early. For computer science students, this means that they will have a better appreciation of how various areas of computer science such as operating systems, software engineering, and many of the more formal courses contribute to system security. For students in other curricula, this early overview of INFOSEC concepts permits them to understand how these ideas are applicable within their own discipline and affords them the opportunity to take more advanced INFOSEC courses as electives.

The second major change to Introduction to Computer Security was the inclusion of extensive laboratory materials to accompany lectures. Although there were occasional demonstrations in class, the course was originally presented with no laboratory component. Now we have developed a set of laboratory exercises and tutorials which complement lecture material. Topics include: passwords, discretionary access controls, mandatory access controls, and use of Pretty Good Privacy (PGP). Student feedback has been very positive as these exercises help to reinforce concepts discussed in lectures and give concrete examples of security implementations. In addition, students become familiar with a range of trusted products and security enhancements to untrusted systems. These include Sun's Trusted Solaris and Wang Federal's XTS 300 system.

The course has been organized into eleven one-week units designed as a logical progression in INFOSEC principles. The prerequisites are: an introductory course on computer organization. It consists of three hours of lecture and two hours of laboratory work per week. We usually give three exams of equal weight during the course and collect approximately six homework and laboratory assignments. The catalog description is quoted here:

This course is concerned with fundamental principles of computer and communications security for modern monolithic and distributed systems. It covers privacy concerns, data secrecy and integrity issues, as well as DoD security policy. Security mechanisms introduced will include access mediation, cryptography, authentication protocols, and multilevel secure systems. Students will be introduced to a broad range of security concerns including both

environmental as well as computational security. Laboratory facilities will be used to introduce students to a variety of security-related technologies including, discretionary access controls in Class C2 systems, mandatory access controls in both low and high assurance systems, identification and authentication protocols, the use of cryptography in distributed systems, and database technology in trusted systems.

With few books to choose from as texts, we elected to use a book that would give an overview of the field [15] and to provide an extensive set of other materials for assigned readings. Because the book had no homework problems, we had to devise all homework sets ourselves. Below is a brief outline of the topics covered in the NPS CISR version of Introduction to Computer Security. The references are to the supplementary reading assigned for each topic. One of the articles [7] is assigned over several weeks, because it covers a number of topics.

- Introduction to Computer Security- Definition, laws, historical perspective.
- Access Control I - Policies, Identification and Authentication, Discretionary Access Control [7]
- Access Control II - Mandatory Access Control and Supporting Policies [7]
- Building Secure Systems I - Design and Implementation concepts that support assurance [3]
- Malicious Software and Intrusion Detection - Trojan Horses, viruses, worms, detecting attacks. [9]
- Certification and Accreditation, Disaster Planning and Recovery, and Risk Analysis - certification and accreditation issues [2]
- Cryptography basics - private key, public key, and hashing schemes
- Cryptographic protocols - key management, voting, digital cash, secret sharing, one time password generation, Digital Signature Standard and Clipper. [10] [19] [21]
- Network Security - special considerations, combining access control and cryptography. [7]
- Network Security in Today's Environment - TCP/IP, Internet and firewalls [5] [8] [20]
- Building Secure Systems II - System evaluation issues [18] [13]

Like the subject it surveys, Introduction to Computer Security is dynamic. Each quarter we review the topics covered as well as the readings to ensure that they remain current and pertinent. We hope that this description of our course will encourage the interested reader to review the course notes which have been included as an appendix and to read some of the articles that we believe are useful supplements to the book.

References

1. OPNAV INSTRUCTION 5239.X, Working Draft, 21 June 1996.
2. Issues in Quantitative versus Qualitative Risk Analysis, Datapro Reports on Information Security, IS20-250-101, McGraw-Hill, Delran, NJ, 1992.
3. Ames, S. H., Gasser, M. and Schell, R. R, Security Kernel Design and Implementation: An Introduction, IEEE Computer, Vol. 16, pp. 14-22, 1983.
4. Anderson, James P, Computer Security Technology Planning Study, Air Force Electronic Systems Division, ESD-TR-73-51, Hanscom AFB, Bedford, MA, 1972. (Also available as Vol. I, DITCAD-758206. Vol. II, DITCAD-772806)
5. Bagwill, R., Carnahan, L, Kuhn, R., Nakassis, A. Ransom, M., Barkley, J., Chang, S., Markovitz, P., Olsen, K., and Wack, J. Security in Open Systems, NIST Special Publication 800-7, ed. Barkley, Computer Systems Technology, U.S. Department of Commerce, National Institute of Standards and Technology.
6. Brinkley, D. L., and Schell, R. R., What Is There to Worry About? An Introduction to the Computer Security Problem, in Information Security: An Integrated Collection of Essays, ed. Abrams and Jajodia and Podell, IEEE Computer Society Press, Los Alamitos, CA, pp. 11-39, 1995.
7. Brinkley, D. L., and Schell, R. R., Concepts and Terminology for Computer Security, in Information Security: An Integrated Collection of Essays, ed. Abrams and Jajodia and Podell, IEEE Computer Society Press, Los Alamitos, CA, pp. 40-97, 1995.
8. Cheswick, W. R.. and Bellovin, S. M, An Evening with Berford In which a Cracker is Lured, Endured, and Studied, Chapter 10 in Firewalls and Internet Security, Addison Wesley, Reading, MA, 1994.
9. Denning, D., Neumann, P., and Parker, D., Social Aspects of Computer Security, in Proceedings 10th National Computer Security Conference, pp. 320-325, September 1987.
10. Denning, D., and Branstad, D., A Taxonomy for Key Escrow Encryption Systems, Comm. A.C.M., Vol 39, p. 34, 1996.
11. Fithen, K., and Fraser, B., CERT Incident Reponse and the Internet, Comm. A.C.M., Vol 37, pp. 108-133, 1994.
12. Landau, S., Kent, S., Brooks, C., Charney, S., Denning, D., Diffie, W., Lauck, A., Miller, D., Neumann, P., and Sobel, D., Crypto Policy Perspectives, Comm. A.C.M., Vol. 37, p. 115, 1994.
13. Lee, T. M. P., A Note on Compartmented Mode: To B2 or Not To B2?, in Proceedings 15th National Computer Security Conference, pp. 448-458, 1992.

14. Lunt, T. F., A Survey of Intrusion Detection Techniques, *Computer and Security*, Vol. 12, pp. 405-418, 1993.
15. Russell, D., and Gangemi, G. T., *Computer Security Basics*, O'Reilly & Associates, Inc., 1991.
16. Saltzer, J. H, and Michael D. Schroeder, M.D., The Protection of Information in Computer Systems, *Proceedings of the IEEE*, Vol. 63, No. 9, pp. 1278-1308, 1975.
17. Schell, Roger R., Computer Security: The Achilles' Heel of the Electronic Air Force, *Air University Review*, January-February, pp. 16-33, 1979.
18. Schell, R. R., and Brinkley, D. L., Evaluation Criteria for Trusted Systems, in *Information Security: An Integrated Collection of Essays*, ed. Abrams and Jajodia and Podell, IEEE Computer Society Press, Los Alamitos, CA, pp. 137-159, 1995.
19. Schneier, B., Cryptography, Security, and the Future, *Comm. A. C. M.*, Vol. 40, p. 138, 1997.
20. Wack, J.P. and Carnahan, L. J., Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls, *NIST Special Publication 800-10*, U.S. Department of Commerce, National Institute of Standards and Technology.
21. Walker, S.T., Lipner, S.B., Ellison, C.M., and Balenson, D.M., Commercial Key Recovery, *Comm. A.C.M.*, Vol. 39, p. 41, 1996.
22. Ware, W., Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security. Rand Corporation, 1970. AD-A076617/0.
23. Weissman, C., Security Controls in the ADEPT-50 Time Sharing System. In *Proceedings of the 1069 AFIPS Fall Joint Computer Conference*, pp. 119-133. AFIPS Press, 1969.

Appendix 1 - Table of Contents

An Introduction to Computer Security.....	A1-1
Access Control I - Identification and Authentication and Discretionary Access Control Policies	A1-29
Access Control II - Mandatory Access Control Policies and Supporting Policies	A1-53
Building Secure Systems I	A1-81
Malicious Software and Intrusion Detection.....	A1-121
Accreditation, Certification, Disaster Planning, and Risk Analysis.....	A1-133
Basics of Cryptography.....	A1-197
Cryptographic Protocols and Applications	A1-241
Network Security	A1-277
Network Security in Today's Environment	A1-317
Building Secure Systems II - System Evaluations	A1-337

Section 1

An Introduction to Computer Security

Course Overview

Sections

1. *Introduction to Computer Security*

- Computer Security definition, laws, historical perspective

2. *Access Control I*

- Identification and Authentication and Discretionary Access Control

3. *Access Control II*

- Mandatory Access Control and Supporting Policies

4. *Building Secure Systems I*

- Design and implementation concepts that support assurance

5. *Malicious Software and Intrusion Detection*

- Trojan Horses, viruses, worms, etc.

6. *Certification and Accreditation, Disaster Planning and Recovery and Risk Analysis*

- Certification and accreditation issues

Course Overview

7. *Basics of Cryptography*

- Private key, public key and hashing schemes

8. *Cryptographic Protocols and Applications*

- Cryptographic protocols for providing secrecy, integrity and authentication

9. *Network Security*

- Special considerations, combining access control and cryptography

10. *Network Security in Today's Environment*

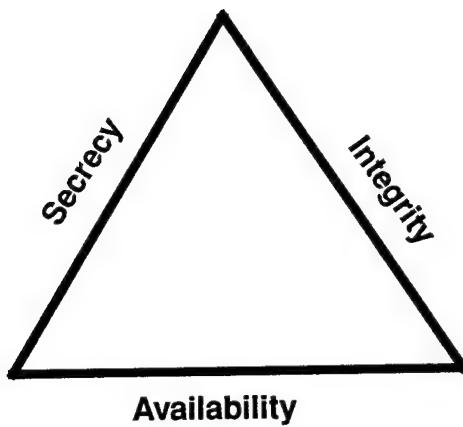
- TCP/IP, Internet and firewalls

11. *Building Secure Systems II*

- System evaluation issues

Aspects of Computer Security

The Golden Triangle of COMPUSEC



Broad definition also includes:

- Physical Security
- Emissions Security
- Personnel Security

Who Should be Concerned?

As a Member of DoD

- You are required to safeguard classified material.
- You are required to safeguard sensitive but unclassified material such as privacy act data.

As a Private Citizen

Questions you should be asking about information stored about you:

- Who has it?
- Who is selling it?

Personal data (age, phone, address, SSN, etc.)

Credit information

Medical information

Purchasing history

Issues for Concern

Why Would Someone Attempt Unauthorized Access?

- Curiosity
- Vandalism
- Financial gain
- Intelligence gathering
- Terrorism
- Warfare

Violations to Data Secrecy

- Wiretaps
- Obtain classified information
- Obtain financial data

Violations to Data Integrity

- Alter bank records
- Alter source e-mail address

Violations Affecting Availability

- Steal Time
- Deny Service

Legislation Addressing Computer Security Issues

Three types of laws address computer security

- Laws about classified and sensitive but unclassified (SBU) data.
- Computer crime laws.
- Laws regarding privacy issues.

Protection of Classified or Sensitive Information

National Security Decision Directive 145

(NSDD 145) - 1984

- Mandated protection of both classified and unclassified sensitive information.
 - Productivity statistics
 - Census Bureau statistics
 - Air traffic control information
 - Health and financial records
- Gave NSA jurisdiction to encourage, advise and assist in the private sector (controversial to say the least).
- Created the System Security Steering Group
 - Secretaries of Defense, State and Treasury
 - Attorney General
 - Director of OMB
 - Director of CIA
- Revised and reissued in 1990 as NSDD 42
 - Scope narrowed to primarily defense related information.

National Telecommunications and Information Systems Security Publication 2

(NTI SSP 2) - 1986

"National Policy on Protection of Sensitive but Unclassified Information in Federal Government Telecommunications and Automated Systems"

Sensitive, but unclassified information is information the disclosure, loss, misuse, alteration, or destruction of which could adversely affect national security or other federal government interests. National security interests are those unclassified matters that relate to the national defense or the foreign relations of the U.S. government. Other government interests are those related, but not limited to the wide range of government or government derived economic, human, financial, industrial, agricultural, technology, and law enforcement information, as well as the privacy or confidentiality of personal or commercial proprietary information provided to the U.S. government by its citizens.

- Applies to all government agencies and contractors.
- Rescinded in March 1987 due to privacy concerns.

Protection of Classified or Sensitive Information

Computer Security Act

Public Law 100-235 (1987)

- Requires every U.S. government computer system that processes sensitive information to have a customized security plan.
- Requires all users of these systems (federal employees or contractors) to receive computer security training.
- Further defines sensitive information as:

"...information whose loss, misuse, unauthorized access to, or modification of could adversely affect the national interest, or the conduct of federal programs, or the privacy to which individuals are entitled to under ... the Privacy Act"
- Gave the Institute of Computer Sciences and Technology (branch of NIST) responsibility for assessing the vulnerability of federal computer systems, for developing standards, and for providing technical assistance as well as developing guidelines for the training of personnel

DoD Directive 5200.28

Security Requirements for Automated Information Systems

- Provides mandatory, minimum AIS security requirements.
- Promotes the use of cost-effective, computer-based security for AISs.
- Applies to classified information, sensitive unclassified information and unclassified information.
- Applies to all AISs (stand-alone systems, communications systems, computer network systems, peripheral devices, embedded computer systems, personal computers, word processors, office automation systems, application and operating systems software and firmware).
- Requires all DoD systems to be accredited.
 - Outlines the accreditation process.
 - Specifies accreditation responsibilities, DAA, ISSO, etc.

Computer Crime

18 U.S. Code 1005 (1948)

- Prohibits making false entries in bank records.

18 U.S. Code 1006 (1948)

- Prohibits making false entries in credit institution records.

18 U.S. Code 1362 (1948)

- Prohibits malicious mischief to government property.

18 U.S. Code 2071 (1948)

- Prohibits concealment, removal, or mutilation of public records.

18 U.S. Code 1343 (1952)

- Prohibits wire fraud using any interstate communications system.

18 U.S. Code 1029 (1984)

- Prohibits fraudulent use of credit cards, passwords, and telephone access codes.

18 U.S. Code 2701 (1986)

- Prohibits unauthorized access to information that's stored electronically.

18 U.S. Code 2778 (1989)

- Prohibits illegal export of software or data controlled by the DoD

18 U.S. Code 2510 (1989)

- Prohibits the illegal export of software or data controlled by the Dept. of Commerce.

Computer Crime

Computer Fraud and Abuse Act

Public Law 99-474 (1986)

- Prohibits unauthorized or fraudulent access to government computers.
- Prohibits access with intent to defraud.
- Prohibits intentional trespassing.
- Applies to computers containing national defense, banking or financial information.
- Establishes penalties.
 - Fine of \$5000 or twice the value of anything obtained.
 - Up to five (5) years in jail.
- Robert T. Morris (first person convicted - 1990)
 - 3 yrs probation, \$10,000 fine, 400 hrs community service
 - Supreme Court refused to hear his case
- Wording vague
 - Must show intent to use information to injure U.S. or to provide advantage to foreign nation.
 - No distinction between those who use computers for hacking, crime or terrorism.

Privacy

U.S. Constitution, Bill of Rights

(1791)

- Fourth Amendment guarantees protection against unreasonable search and seizure.
(Note: The Constitution does not explicitly guarantee an individuals right to privacy!)

Privacy Act

Public Law 93-579 (1974)

- Requires U.S. government to:
 - safeguard personal data processed by federal agency computer systems.
 - provide ways for individuals to find out what information is being recorded on them and a means to correct inaccuracies.

Right of Financial Privacy Act

(1978)

- Establishes that a depositor's bank accounts are private
- Can be accessed only by court order and proper notification

Electronic Funds Transfer Act

(1979)

- Protects the privacy of transmission of funds using electronic funds transfer (EFT)

Privacy

Electronic Communications Act

(1986)

- Prohibits unauthorized interception of communications regardless of how transmission takes place:
 - wire
 - radio
 - electromagnetic
 - photo-electric
 - photo-optical

Computer Matching and Privacy Protection Act

U.S. Code 552a (1988)

- Protects against privacy violations due to information matching policies of the federal government.

Early Computer Security Efforts

1950's

- Stand alone systems using physical security for systems and terminals.
- Development of first TEMPEST standard.
- Establishment of U.S. Communications Security (COMSEC) Board.

1960's

- Beginning of the age of Computer Security.
- Timesharing systems (multiple users at the same time) and remote terminals created new problems.
- DoD launched first study of threats to DoD computers (1967).
 - Assembled task force under Advanced Research Projects Agency (ARPA).
 - Published findings in 1970 Security Controls for Computer Systems

Early Computer Security Efforts

1970's

Security Requirements for Automatic Data Processing (ADP) Systems - issued by DoD in 1972

“Classified material contained in an ADP system shall be safeguarded by the continuous employment of protective features in the system's hardware and software design and configuration.”

- Tiger Teams were used to test vendor claims of computer security features by attempting to break into the vendor's system.
 - The Tiger Teams showed overwhelming success at breaking in.
 - Patching techniques were used to shore up a system's weaknesses.
 - After patching, systems were still penetratable.
 - The patch and penetrate scheme shown to be inherently flawed. The concept of a Reference Monitor (Security Kernel) is spawned.
 - Most Tiger Teams were sponsored by DoD.
- IBM spends \$40 Million to address computer security issues.
- First mathematical model of a multi-level security policy.
 - Developed by David Bell and Leonard LaPadula.
 - Central to development of computer security standards.
 - Laid groundwork for later models.
- Development of first security kernel.
 - USAF develops security Kernel for Multics system.
- Other kernels under development.
 - Mitre's DEC PDP-11/45
 - UCLA's Data Secure UNIX PDP-11/70

Early NBS and NSA (NCSC) Involvement

National Bureau of Standards (NBS)

(National Institute of Standards and Technology - NIST)

- 1968 - NBS performed an initial study to evaluate the government's computer security needs.
- 1972 - NBS sponsored a conference on computer security in collaboration with ACM.
- 1973 - NBS initiated program aimed at researching development standards for computer security.
- 1977 - NBS began a series of Invitational Workshops dedicated to the Audit and Evaluation of Computer Systems.

Conclusion in NBS report from 1977 workshop

“...The point is that internal control mechanisms of current operating systems have too low integrity for them to... effectively isolate a user on the system from data that is at a 'higher' security level than he is trusted... to deal with.”

National Computer Security Center

- 1980 - Director of NSA assigned responsibility for trusted information security products.

Response to NBS workshop and public seminars on the DoD Computer Security Initiative.

- 1981 - DoD Computer Security Center (CSC) was established
- 1985 - CSC becomes NCSC and assumes scope of responsibility broadens.
- 1985 - Communications and computer security merge under Deputy Directorate for Information Security Systems (INFOSEC).

Tasking of the NCSC

- Encourage the widespread availability of trusted computer systems.
- Evaluate the technical protection capabilities of industry and government developed systems
- Provide technical support of government and industry groups engaged in computer security research and development.
- Develop technical criteria for the evaluation of computer systems.
- Evaluate commercial systems.
- Conduct and sponsor research in computer and network security technology.
- Develop and provide access to verification and analysis tools used to develop and test secure computer systems.
- Conduct training in the areas of computer security.
- Disseminate computer security information to other branches of the federal government and to industry.

Why Is Computer Security Difficult?

Some Factors

- Most managers are unaware of the value of their own computing resources.
- Fear of damage to public image.
- Legal definitions are often vague or non-existent.
- Legal prosecution is difficult:
 - Criminal must be traced.
 - No 'hard' evidence.
 - Hard to pin a value to data.
 - "No fingerprints" mentality.
 - Criminals viewed as just curious intellectuals.
- Computer Criminals do not fit a stereotype.
- The Law and Ethics are often unclear.

Computer Security Problems and Crimes

Computer Security Problems

- Most loss or damage is not malicious
 - Ignorance of existing policies.
 - Ignorance of the system on which they work.
- Accidents
 - Anyone can make a mistake!

Computer Security Crimes

Amateurs

- Temptation is there if access is available.
 - You wouldn't ask a stranger to hold your wallet while you went around the corner to move your car.
- Disgruntled employees
 - Oh Yeah! I'll show you!

Crackers and Hackers

- Often the challenge or Curiosity
 - West German group (Cliff Stoll)
 - Desert Shield / Desert Storm

Corporate Raiders

- Trade Secrets
- Inside Information
- Financial predictions

Foreign Intelligence

- West German group (Cliff Stoll)
- Desert Shield / Desert Storm

Terrorists

- No major incidents have occurred yet!
 - This is a potential nightmare waiting to happen.
 - Potential Economic disaster.

Categories of Computer Misuse

Human Error

- Hard to control

Abuse of Authority

- White collar crime

Direct Probing

- Rattling doorknobs

Probing With Malicious Software

- Trojan Horses

Direct Penetration

- Exploiting system bugs

Subversion of Mechanism

- Trap doors

Is There A Threat?

Bank Theft (1984)

- Branch manager netted \$25 million!

HBO Attack (1986)

- Captain Midnight overpowered HBO uplink.
- Part-time uplink operator.
- Displayed brief message to viewers.

Chaos Club

- West German Computer Club.
- In 1987 announced that it had successfully penetrated a United States Government Computer (NASA's).
- Able to store and manipulate information on SDI.
- NASA was unaware of penetration until messages started appearing on the system.
- NASA initially reported no damage.
- Virus later found on system which may have originated during the initial break-in.

Cliff Stoll and the KGB

- West German crackers tried to break in to over 450 computers (1987).
- 30 successful attempts.
- Looking for NBC related information to sell to KGB.
- First prosecution for Computer espionage.

Airline Computers (1988)

- A major airline discovered its reservation and ticketing system had been penetrated.
- Bogus reservations had been made.
- Illegal tickets issued.

Internet Worm (1988)

- Affected Sun and VAX systems.
- 2100-2600 systems affected.
- Culprit: Robert T. Morris, a Cornell graduate student.

Is There A Threat?

Friday 13th Virus (1988)

- Threatened to erase the hard disks of financial, research and administrative computers.
- Originated at Hebrew University in Jerusalem.

Virus Flambe (1988)

- Infected a computer consulting firm.
- Altered scan rate of IBM monitors.
- Monitor burst into flames.

Satellite Positioning System (1989)

- 14 year old boy using Apple computer:
 - Cracked Air Force SPS.
 - Dialed unauthorized long-distance access codes.
 - Browsed through file of 200 businesses.

Desert Storm/Desert Shield (1990)

- 40 known attempts (6 confirmed successful)!
- How many unknown attempts/successes

Airline traffic control system (Arorua IL) (1995)

- Air traffic delayed for several hours!

Word Macro Virus (1995)

- Currently the most prevalent virus.

ActiveX Trojan Horse (1997)

- Transfers funds from your bank account to the hacker's account.

Erotic Photograph Trojan Horse (1997)

- Reconnects your system through a phone number in Moldova.

There is a problem and it's getting worse!

Economic Damage from the Internet Worm

INDIRECT COSTS		
	<i>Lost Machine Time</i>	<i>Lost Access</i>
Machine hours unable to access network	2,076,880	
User hours unable to access network	8,307,520	
Burdened cost per hour	\$20	\$3
COST	\$41,537,600	\$24,922,560
DIRECT COSTS		
	<i>Programmer Time</i>	<i>Admin. Time</i>
Shutdown, monitor and reboot 42,700 machines	64,050	1,000
Initial problem analysis 12,400 machines	49,600	11,000
Identify, isolate, remove, clean, return to operation (6,200)	74,400	2,000
Reinfection, removal from network, shutdown analysis, monitor	62,000	12,000
Create patch, debug, install, test, check-out, monitor and implement	62,000	18,000
Analyze worm, disassemble, document (at each of 1200 networks)	192,000	22,000
Install fix on all UNIX systems, test, checkout and monitor	105,000	6,000
Residual checkup, tech communications conferencing and ripple events	187,000	264,000
TOTAL HOURS	796,050	336,000
Hourly Rate	\$22	\$42.50
COST	\$17,513,100	\$14,280,000
TOTAL COSTS: \$98,253,260		

Information Warfare

Overview

- Practically all of today's vehicles, ships and aircraft use control devices, communications systems and weapons systems that use computer systems, in one form or another. These systems, as in all information systems, rely upon the integrity of programs and the data which they input and output. Therefore, it is reasonable to assume that any attack on the data and programs may render these systems useless.

Hard kill, Soft kill or Imperceptible Degradation

- We frequently think in terms of complete system kills, either from a direct missile hit to the platform (hard kill) or to supporting sensor systems (soft kill) which the platform relies upon for its navigation, communication or targeting. The benefit of course, is that we can achieve total destruction of the system or render it completely useless; however, the major disadvantage is that our adversary is aware of the loss of their system and may engage alternate systems.
- Less obvious is the advantage that can be achieved through imperceptible degradation of the targeted information system. If we can reduce the measure of effectiveness of the system, often referred to as probability of kill (PK), in such a fashion that our adversary is unaware and places his faith in the system then we can effectively increase our force multiplier.
- There are a number of tradition battlefield models which have proven reasonably accurate in measuring overall force effectiveness on actual battlefields. Modification of these models to reflect system degradation to the enemy's control, communication and weapons systems have shown that viruses and worms can have a significant effect on battlefield results.

Insertion Techniques

- Insertion of viral code may take place during the manufacturing or distribution process. However, it is possible that the code could be inserted by field operatives during a time of crisis.
- The code could be triggered remotely or by use of logic bombs that render the system completely ineffective or slightly degrade system performance. The code obviously needs be hardware/software specific; that is, the code must be written to target a specific communications or weapons system.

Information Warfare

Battle Damage Assessment (BDA)

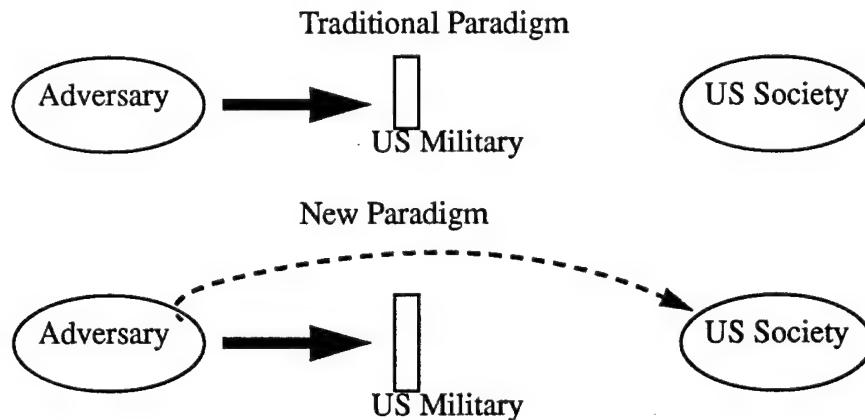
- Battle damage assessment for complete systems kill is challenging enough and frequently relies upon sophisticated surveillance systems to provide the battlefield commander with accurate information. Battle damage assessment for systems which have their PK perturbed slightly cannot be immediately assessed and requires analysis of battlefield results over prolonged periods.
- This factor makes it difficult for commanders to assess the overall effectiveness of such warfare techniques during the fog of war and as a result they may actually be unaware that it has achieved an advantage in their favor.

IW Difficulties

- What constitutes an act of war?
- What is the correct response?
- The civilian sector is not prepared to deal with attacks.

IW issues:

- IW attacks can be perpetrated with very little resources.
- Presents a different attack paradigm.



This page is intentionally blank.

This page is intentionally blank.

Section 2

Access Control I

*Identification and Authentication and
Discretionary Access Control Policies*

Access Control Policies

Security Policy:

- Generally speaking, a *security policy* describes how people may access documents or other information.
- A computer's version of a security policy consists of a precise set of rules for determining authorization as a basis for making access control decisions.
- This section and the following section present several security policies that are commonly implemented in computer systems.
- Policies presented include:
 - Access to systems based upon user identification.
 - Access to objects (such as files, directories, etc.) based upon user identification, where owners of objects can, at their discretion, grant access to other users.
 - Access to objects (such as files, directories, etc.) based upon the clearance level of the user.

System Access Control

Controlling Access to the System Physically

Guards

- need at least 4 for 24 hour coverage
- must recognize someone, or token
- no record of access

Locks

- cheaper than a guard
- no record of access

Identification and Authentication (I and A)

Controlling Access to the System Using Identification and Authentication

Two Step Process

Identification

- Telling the system who you are.

Authentication

- Proving to the system that you are who you say you are.

Three classic ways of establishing proof.

- Something you know.
- Something you have.
- Something you are.

I and A Benefits

- Can provide a complete log of access and attempted accesses.
- Access privileges granted/removed quickly

I and A

Passwords

- Something you know.
- Agreed upon code words entered by user.
- Subject to:
 - Loss
 - Disclosure
 - Attack

Attacks on Passwords

- Brute force attack.
 - Try all words.
- Probable password attack.
 - Try short words.
 - Try common words.
- Probable user password attack.
 - Family names.
 - Birth dates.

Password File

- Conventional encryption.
 - Enter password.
 - Decrypt stored password from table.
 - Compare passwords.
- One way cipher.
 - Enter password.
 - Encrypt password.
 - Compare to encrypted password.

Attacks Using Password File

- Readable password file.
- Backup tapes.

Guessing Passwords

Password Space:

- The password space is the set of all passwords.
- The size of a password space is determined by:
 - The length of passwords, denoted by L .
 - The size of the password alphabet, denoted by A .
 - If passwords only consist of lower case letters, $A = 26$.
 - If passwords consist of lower and upper case letters and digits, $A = 62$.
- The size of the password space is A^L .

L	26^L	52^L	62^L
4	4.57×10^5	7.31×10^6	1.47×10^7
6	3.09×10^8	1.98×10^{10}	5.68×10^{10}
8	2.09×10^{11}	5.34×10^{13}	2.18×10^{14}
10	1.41×10^{14}	1.44×10^{17}	8.39×10^{17}

Exhaustively trying all passwords:

- On the average, you will need to try half of them.
- If an intruder (using a computer) tries 1 password each second, they can try 60 passwords a minute, or 86,400 passwords a day.
- If passwords are of length 6 and consist of lower case letters, it will take 60 months, on the average.
- If an English word is used as a password, the problem is greatly simplified. There are only 5000 8-letter English words. The intruder can guess one of these in 42 minutes, on average.
- If the intruder steals an encrypted password file and the encryption software, it takes only 10^{-6} seconds to check whether an encrypted string is one of the encrypted passwords.
- Thus, potential passwords can be tested 1,000,000 times faster.
- A 6-letter password can be guessed in 155 seconds, on average.

Internet Worm Password Guesses:

- The following list shows passwords that the Internet worm tried..

Passwords Tried by the Internet Worm

aaa	beater	comrades	engine	golfer
academia	beauty	computer	engineer	gorgeous
aerobics	beethoven	condo	enterprise	gorges
airplane	beloved	cookie	enzyme	gosling
albany	benz	cooper	erastz	gouge
albatross	beowulf	cornelius	establish	graham
albert	berkeley	couscous	estate	gryphon
alex	berliner	creation	euclid	guest
alexander	beryl	creosote	evelyn	guitar
algebra	beverly	cretin	extension	gumption
aliases	bicameral	daemon	fairway	guntis
alphabet	bob	dancer	felicia	hacker
ama	brenda	daniel	fender	hamlet
amorphous	brian	danny	fermat	handily
analog	bridget	dave	fidelity	happening
anchor	broadway	december	finite	harmony
andromache	bumbling	defoe	fishers	harold
animals	burgess	deluge	flakes	harvey
answer	campanile	desperate	float	hebrides
anthropegenic	cantor	develop	flower	heinlein
anvils	cardinal	dieter	flowers	hello
anything	carmen	digital	foolproof	help
aria	carolina	discovery	football	herbert
ariadne	caroline	disney	foresight	hiawatha
arrow	cascades	dog	format	hibernia
arthur	castle	drought	forsythe	honey
athena	cat	duncan	fouier	horse
atmosphere	cayuga	eager	fred	horus
aztecs	celtics	easier	friend	hutchins
azure	cerulean	edges	frighten	imbroglio
bacchus	change	edinburgh	fun	imperial
bailey	charles	edwin	fungible	include
banana	charming	edwina	gabriel	inges
bananas	charon	egghead	gardner	inna
bandit	chester	eiderdown	garfield	innocuous
banks	cigar	eileen	gauss	irishman
barber	classic	einstein	george	isis
baritone	clusters	elephant	gertrude	japan
bass	coffee	elizabeth	ginger	jessica
bassoon	coke	ellen	glacier	jester
batman	collins	emerald	gnu	jixian

Passwords Tried by the Internet Worm

johnny	mike	philip	rules	success
joseph	minimum	phoenix	ruth	summer
joshua	minsky	pierre	sal	super
judith	moguls	pizza	saxon	superstage
juggle	moose	plover	scamper	support
julia	morley	plymouth	scheme	supported
kathleen	mozart	polynomial	scott	surfer
kermit	nancy	pondering	scotty	suzanne
kernel	napoleon	pork	secret	swearer
kirkland	nepenthe	poster	sensor	symmetry
knight	ness	praise	serenity	tangerine
ladle	network	precious	sharks	tape
lambda	newton	prelude	sharon	target
lamination	next	prince	sheffield	tarragon
larkin	noxious	princeton	sheldon	taylor
larry	nutrition	protect	shiva	telephone
lazurus	nyquist	protozoa	shivers	temptation
lebesque	oceanography	pumpkin	shuttle	thailand
lee	ocelot	puneet	signature	tiger
leland	olivetti	puppet	simon	toggle
leroy	olivia	rabbit	simple	tomato
lewis	oracle	rachmaninoff	singer	topography
light	orca	rainbow	single	tortoise
lisa	orwell	raindrop	smile	toyota
louis	osiris	raleigh	smiles	trails
lynne	outlaw	random	smooch	trivial
macintosh	oxford	rascal	smother	trombone
mack	pacific	really	snatch	tubas
maggot	painless	rebecca	snoopy	tuttle
magic	pakistan	remote	soap	umesh
malcolm	pam	rick	socrates	unhappy
mark	papers	ripple	sossina	unicorn
markus	password	robotics	sparrows	unknown
marty	patricia	rochester	spit	urchin
marvin	penguin	rolex	spring	utility
master	peoria	romano	springer	vasant
maurice	percolate	ronald	squires	vertigo
mellon	persimmon	rosebud	strangle	vicky
merlin	persona	rosemary	stratford	village
mets	pete	roses	stuttgart	virginia
michael	peter	ruben	subway	warren

Passwords Tried by the Internet Worm

water	will	wisconsin	yacov	zimmerman
weenie	william	wizard	yang	
whatnot	williamsburg	wombat	yellowstone	
whiting	willie	woodwind	yosemite	
whitney	winston	wormwood	zap	

Password Issues

Password Issues

- Use more than just A-Z.
- Use a password of at least 6-characters
- Avoid actual names or words.
- Choose an unlikely password.
- Change your password regularly.
- Don't write it down.
- Don't tell it to someone else.
- Avoid shoulder-hangers.

Implementation Issues:

- System may actually give away information.
 - Which part of login is incorrect.
 - Which system is being accessed.
- Limit access attempts.
- Enforce password time limits.
- Employ terminal restrictions
- Employ password checking programs.
 - Proactive checkers are best.
 - Ensures adequate password length.
 - Ensures adequate password alphabet (forces the inclusion of capital letters, punctuation, or numbers).
 - Avoids the use of English words.

Authentication Devices

Tokens and Smart cards

- Something you have.
- A token is an object which authenticates its possessor.
- Must be unforgeable and unique.
- Not foolproof since it may be lost or stolen.
- Smart card may compute the response to challenge.
- Smart card may perform encryption.

Personal Characteristic Recognition (Biometric Devices)

- Something you are.
- Retinal scanners.
- Palm/fingerprints.
- Voice pattern recognition.
- Difficult for imposter to duplicate.

Challenge and Response Systems

- Something you have and something you know.
- Passwords are in the clear from time of entry until accepted by host.
- Normal passwords are static.
- Challenge and reply systems create a pseudo one time password system.
- Passwords become dynamic.
- To ensure security:
 - Encryption keys should be changed regularly.
 - Algorithms should be changed occasionally.
- Challenge and reply systems are most appropriate for host-to-host communications because of the computing power available.
- This method affords authentication and identification as well as eliminates the replay problem.

Modem Issues

Automatic Call-Back

- Internal table must be well protected.
- This same technique can be used between two hosts that wish to communicate.

Steps

- User dials a computer system.
- User identifies himself/herself to system.
- System breaks communication.
- System consults internal table.
- System calls back at predetermined telephone number.
- If number specified by user not one of those listed in the computer's directory then a warning is issued to security officer.

Silent Modem

- Carrier tone is suppressed until caller sends the first tone.
- Does not reveal that the telephone line is a modem line.
- No real protection, only forces intruder to take a second step.
- Prevents a computer from dialing randomly in search of another computer.

Login Spoofing

Problem:

- A password grabbing program is malicious software that is left running on a terminal that mimics the normal login prompt.
- After a user enters a login name and password, the program records the name and password and displays the normal incorrect password message and exits.
- The correct system login prompt is displayed and the user logs in again, this time without further problems.
- However, the person that left the spoofing program running can retrieve the login name and password and login under an assumed identity.
- This type of program is a type of Trojan Horse program. Specifically, it is a "spoofing" Trojan Horse program. It is also called a "password grabber".

Solution:

- *The Trusted Path*
- An unforgeable link between the terminal and the system.
- When the trusted path is invoked, all user processes to a terminal are killed and the system trusted path screen or menu is displayed.
- It provides a means where the user can be sure that they are communicating with the REAL system.
- Before logging in, users **ALWAYS** invoke the trusted path.
- All password management functions, like changing passwords, should use the trusted path.
- As we will see in other sections, other trusted functions should use the trusted path too.

Note:

- Passwords and biometric devices are **ONLY** good for authenticating the user to the system.
- A *trusted path* is required to authenticate the system to the user.
- I and A consists of both identifying and authenticating the user to the system and identifying and authenticating the system to the user.

Data Access Control

Discretionary Access Control (DAC) is a data access control policy that allows users to grant or deny other users access to their files.

Common implementations

- Permission Bits
- Password Schemes
- Capability Lists
- Access Control Lists (ACLs)

DAC

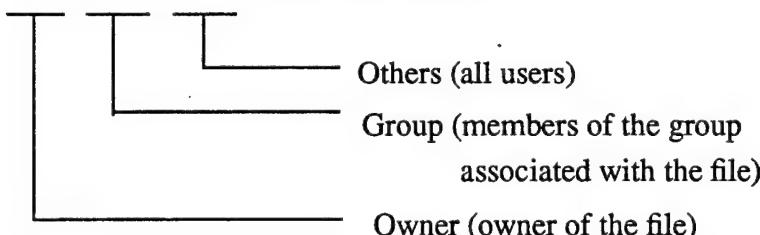
Permission Bits

- Used by Unix, VMS and other systems.
- A user is specified as the owner of each file or directory.
- Each file or directory is associated with a group.
- At any specific time each user is associated with a group.

Example

r w e **r w e** **r w e** bits specifying Read, Write or Execute permission

file1	r w -	r w -	---	Alice (file owner)
file2	r - e	r - e	r - e	Bob (file owner)
file3	r - -	r - -	r - -	Alice (file owner)



Drawbacks

- Insufficient granularity (how does Alice give ONLY Bob read access to file1?).
- Deny access to a single user? No.

DAC

Passwords for file / directory access

- A single password for every file.

Example

- file1 password1
- file2 password2
- file3 password3

Drawbacks

- Loss - forgotten.
- Disclosure - loose lips; requires reprotecting the file.
- Revocation - password must be changed and all legitimate users must be notified.
- System Administration nightmare, too many passwords.

DAC

Capability Lists

General Schema:

- Every object has a unique owner.
- Owner possesses major access rights.
- Owner may declare who has access.
- Owner may revoke access.
- One capability list per user.
- Names all objects user is allowed access to.
- Lists maintained by OS.
- Users cannot access lists directly.

Capability Lists

Example Capability Lists:

Alice's list of capabilities

- file1 (Owner, Read, Write)
- file2 (Read)
- file3 (Execute)

Bob's list of capabilities

- file2 (Owner, Read, Write)
- file5 (Execute)

Trent's list of capabilities

- file3 (Owner, Execute)
- file6 (Read)

Difficulties with Capability Lists Schema:

- Management of large/many lists.
- Revocation of access - must search all user lists to determine if object is on that user's list.

Access Control Lists

General Schema:

- One list for each object.
- Shows all users who have access.
- Shows what access each user has.
- Generally, specifies access based on users and groups.
- Generally, wildcard values are supported to simplify administration.
- Entries are generally listed in order from most specific to least specific and are interpreted in a manner that supports a desired policy. One such policy might be use specific rights over wildcard rights.

Access Control Lists

Example Access Control Lists:

File Alpha

Jones.crypto	rew
Green.*	n
*.crypto	re
.	r

File Beta

Smith.druid	r
.	n

In this example:

- User Jones in group crypto has rew access to file Alpha.
- User Green does not have access to file Alpha.
- All users in group crypto (with the exception of Green) have re access to file Alpha.
- All users, other than Green, have r access to Alpha.
- User Smith in group druid has r access to file Beta.
- No other users have any access to file Beta.

Drawbacks

- Requires a more complicated implementation than permission bits.

DAC Weakness

Suppose you have a system that:

- correctly enforces an I and A policy,
- correctly enforces a DAC policy,
- stores both Unclassified and Secret information, and
- has both Unclassified and Secret users.
- Also suppose that all Secret users act in accordance with procedures for handling classified information (i.e., they do not set access permissions on files containing Secret information such that Unclassified users can view them).

Question: What can go wrong?

Answer: Malicious software.

DAC Weakness

Consider the following scenario:

- An unclassified user, Ivan, brings a great Star Trek game into work. The game becomes very popular. Unbeknownst to users the program surreptitiously copies user's files into Ivan's directories with permissions such that Ivan can read them. This type of program is called a Trojan Horse program. It performs a useful function so that users will use it, but it secretly performs other actions.
- How does the program do this? When Alice, a Secret user, runs programs, those programs (text editors, etc.) are able to access all files accessible by Alice, because those programs are running on behalf of Alice.
- When Alice runs the Star Trek program, it too runs on her behalf and can access all files accessible by Alice. Thus, the game program can read all files readable by Alice and make a copies of them into Ivan's directory with permissions on the files set such that they are readable by Ivan.

The gist is, when Alice runs the game program (or any malicious software) it can do any thing that Alice can do.

Conclusion:

***DAC mechanisms have an inherent weakness.
They are vulnerable to Trojan Horse attacks.***

DAC Weakness

How great is the threat of malicious software?

Consider the following points:

- How much software on your own system did you write?
- How much software on your system can you absolutely vouch for?
- More and more software is written overseas these days.
- It only takes one bad engineer in a group of a thousand good engineers to embed a Trojan Horse in a product.
- If you store information that is worth stealing, the Trojan Horse attack is very attractive
- Are you running a browser that downloads and executes Java applets?

Note:

The users act in accordance with the security policy, it is software that is malicious.

Want to know more?

- A Guide to Understanding Discretionary Access Control in Trusted Systems, NCSC-TG-003

Section 3

Access Control II

*Mandatory Access Control Policies and
Supporting Policies*

Mandatory Access Control Policies (MAC)

Why Do We Need a MAC Policy?

- From Section 2, we know that DAC policies inherently cannot prevent a malicious software (Trojan Horse) attack.
- We need a policy that does address the malicious software problem.
- A MAC Policy is such a policy.

A Mandatory Access Control policy is a policy in which people do not have control over the authorization of people to information.

- Note how this policy differs from a DAC policy.

Within some universe of discourse Mandatory Policies are:

- Global - sensitivity of information does not change relative to its "location" in the system
- Persistent- sensitivity of information does not change from time to time
 - does not state that information is TS on MWF but only C the remaining days of the week

Example MAC Policy

- Military Security Policy

Mandatory Access Control Policies (MAC)

Mandatory Access Control Policy Definitions

Access Class

- User - Clearance
- Information - Sensitivity
- Clearance and Sensitivity can be mapped to system attributes call *Access Classes*.

Object

- Any passive entity that contains information.
- For the time being, think of this as a file.

Subject

- Active entities operating on behalf of users.
- For the time being, think of this as being associated with a process.

In an implementation of a MAC policy

- Each subject has a label (or access class).
- Each object has a label (or access class).
- The ability of a subject to access an object is based upon a comparison of the subject's label and the object's label.
- Two labels are compared using the "dominance" operator " \geq ".
 - I.e., if label A dominates label B, we write $A \geq B$.
- As an example, consider the set of military classification levels {Top Secret, Secret, Confidential, Unclassified}. Where:
 - Top Secret \geq Secret
 - Top Secret \geq Confidential
 - Top Secret \geq Unclassified
 - Secret \geq Confidential
 - etc.
- Technically Top Secret \geq Top Secret , Secret \geq Secret , etc.

Note

- Object labels and subject labels are a requirement of MAC policy implementations.

Bell and LaPadula Model (BLP)

Bell and LaPadula Model Facts

- The Bell and LaPadula Model is a mathematical description of the DoD Security Policy (a later section discusses the need to have a mathematical description).
- The Bell and LaPadula Model specifies read and write access between a subject and an object based upon the dominance relationship between the subject's label (or access class) and the object's label (or access class).
- The Bell and LaPadula Model is the most common model for MAC policies.
- Applies only to secrecy (not integrity) of information.
- It includes both discretionary and mandatory access rules
 - Both checks are made upon request for access.
 - We will only look at the MAC aspects of the model since we are using the model to demonstrate a MAC policy.

BLP Mandatory access control

- Lets S be the set of all subjects in a system and O be the set of all objects in a system.
- For each subject s in S there exists a label or access class for s called $C(s)$.
- For each subject o in O there exists a label or access class for o called $C(o)$.

Bell and LaPadula Model

Basic Properties of Bell-LaPadula Model

Simple Security Property

A subject s may have read access to an object o only if
 $C(s) \geq C(o)$

(You shall only view objects which are classified at the same level or lower than the level for which you are cleared)

** - Property*

Also called ***Confinement Property***

A subject s may have write access to an object o only if
 $C(s) \leq C(o)$

(You shall not talk to anyone who is cleared at a level below you)

The first property (The Simple Security Property) is:

- The normal "no read up" policy where
 - Secret users can read Secret, Confidential and Unclassified information (read down allowed)
 - but Secret users cannot read Top Secret (no read up)

The second property (the *-Property, pronounced 'Star Property') is required to prevent malicious software from writing down.

Bell and LaPadula Model

Why the *-Property is needed

- Recall the Star Trek game that contained a Trojan Horse program. If a Secret user uses the program on a system that **does not enforce the *-Property**, the Trojan Horse could read Secret files and write them to Unclassified files, where Ivan (the person who wrote the Star Trek game) (who is an Unclassified user) can read them.
- If, however, a system enforces the *-Property, a Trojan horse cannot write down.

Thus:

- In a computer system, a mandatory policy can protect information in objects from unauthorized access
even in the face of malicious software.

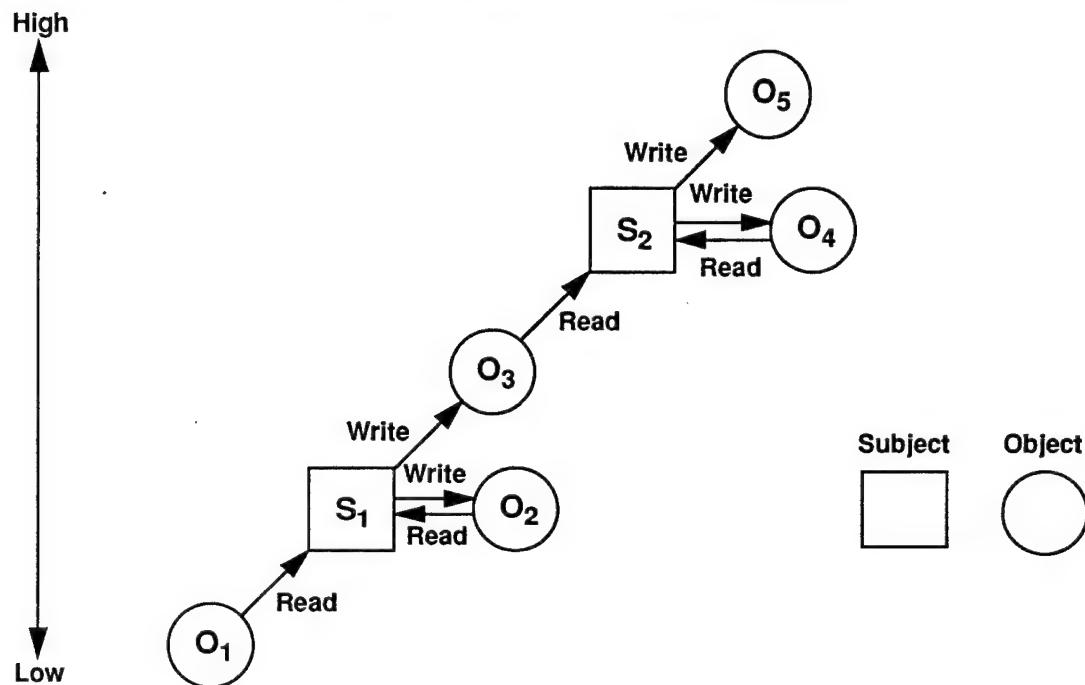
Restatement of the BLP rules:

- No read up.
- No write down.

Bell and LaPadula Model

The BLP Model is often described in terms of secure information flows. The Figure below shows such a flow diagram. This is another way of saying that there is "no read up" and "no write down."

Secure Flow of Information (B-LP)



As indicated by the diagram, a subject can only both read and write an object if the object has the same access class value as the subject.

BLP Example

Consider the following objects and subjects:

- File1 has an access class value of Secret.
- File2 has an access class value of Confidential.
- File3 has an access class value of Top Secret.

- Subject1 has an access class value of Top Secret.
- Subject2 has an access class value of Confidential.

Under the BLP Model the following accesses are allowed:

- Subject1 can read File1, File2 and File3.
- Subject1 can write only File3.

- Subject2 can read File2.
- Subject2 can write File1, File2 and File3.

Can an Unclassified user blindly write to Secret?

- Yes. The model allows it, but most implementations prohibit arbitrary blind write ups.

MAC Issues

Question:

- How does Alice, a Secret user, write information to an Unclassified file?

Answer:

- Systems that support MAC policies, must also support the notion of a *session level*.
- When a user logs on they request a session level, which can be any level up to their clearance level.
- If Alice logs on and requests a session level of Secret, a Secret level subject is created on her behalf. This subject can read files at or below Secret and can write files at or above Secret.
- While Alice is logged in, she can re-negotiate a new session level to any other level that she is allowed to operate at. This means if she needs to write an Unclassified file, she must negotiate an Unclassified session.
- Session negotiation should use the trusted path.

Question:

Who puts the access class label values on objects (files)?

Answer:

- When an object (a file) is created (e.g., with a text editor), its access class value is specified as part of the creation process.
- When files are imported into a system (off a floppy disk, from the network, etc.), they are labelled appropriately.
 - If a file is downloaded from an Unclassified network, it is labelled as Unclassified.
 - If a file is downloaded from a Secret network, it is labelled Secret.
 - If a file is imported off an Unclassified Floppy Disk, it is labelled as Unclassified.
 - If a floppy disk contains multilevel data (files at different access class values), then the files on the disk are labelled accordingly and when they are imported into a system the system label value is made the same as the label value of the file on the disk.

Observation:

- Need a consistent set of machinable labels for heterogeneous systems.

Compartments

Use of compartments

- MAC policies often use Compartments in conjunction with access class levels.
- The term category is sometimes used in place of the word compartment.
- Access class levels generally refer to values that are hierarchically ordered with respect to a dominance operator.
- E.g., $\text{Top Secret} \geq \text{Secret}$, $\text{Secret} \geq \text{Confidential}$, $\text{Confidential} \geq \text{Unclassified}$
- Compartments are not hierarchically ordered values.
- Compartments are set elements where dominance is determined by whether or not a set of compartments is a subset of another set of compartments.

Example:

- Consider a situation where compartment names are fruits.
 - If $A = \{\text{apples, peaches, apricots}\}$ and
 - $B = \{\text{peaches, apricots}\}$ then
 - $A \geq B$ because B is a subset of A .

It is possible to have two sets of compartments C and D , such that C does not dominate D and D does not dominate C .

Example:

- Non-comparable sets of compartments:
 - $C = \{\text{oranges, apples, peaches}\}$ and
 - $D = \{\text{oranges, peaches, bananas}\}$.

Beware:

- Often DoD usage of the term "need-to-know" to refers to the use of compartments and non-DoD literature often uses the term "need-to-know" to refer to DAC.

Compartments and Levels

Examples:

- Levels = Top Secret, Secret, Confidential
- Compartments = Crypto, Nuclear, Biological, Red, Green
 - Alice is logged in at a session level of "Secret-Nuclear, Red"
 - Tim is logged in at a session level of "Confidential-Crypto, Nuclear, Biological"
 - Anne is logged in at a session level of "Top Secret-Green"
- File1 has class "Secret-Green"
- File2 has class "Secret-Red, Green"
- File3 has class "Confidential-Red"
- File4 has class "Top Secret-Green"
- File5 has class "Secret-Nuclear, Red, Green"

Assuming the BLP model of read and write access (where write up is allowed), who has read access to which files and write access to which files?

Integrity

Note:

- The term integrity is used in two ways in the context of computer security.
- Program or execution integrity refers to a system's ability to provide protected domains of execution.
- Data integrity refers to keeping data free from unauthorized modification.

Secrecy versus Integrity

- Recall from the "Golden Triangle" slide that secrecy and data integrity concerns are distinct.
- Secrecy concerns the prevention of unauthorized disclosure of data or information.
- To re-enforce the orthogonal nature of these concepts, provide examples of the four types of data labeled in the table below:

	High Integrity	Low Integrity
High Secrecy		
Low Secrecy		

Question:

- Where does data integrity fit into a MAC scheme that enforces the BLP Model?

Answer:

- Nowhere.

Biba Integrity Model

Biba Model

- In addition to enforcing a policy for secrecy, we would like systems to enforce a mandatory policy for data integrity too.
- The Biba Integrity Model addresses the unauthorized modification problem by restricting read and write accesses.
- Uses integrity levels and integrity compartments much like sensitivity levels and sensitivity compartments.
- For each subject s in S and each object o in O :
 - Fixed integrity classes $I(s)$ and $I(o)$
- A high integrity file is one whose contents are created by high-integrity processes.
 - The properties guarantee that the high-integrity file cannot be contaminated by information from low-integrity processes.
 - The high-integrity process that writes the file cannot be subverted by low integrity processes or data.
 - The integrity class label on a file guarantees that the contents came only from sources of at least that degree of integrity.

Biba Integrity Model

Basic Properties of Biba Model

Simple Integrity Property

A subject s can modify (have write access to) object o , only if
 $I(s) \geq I(o)$.

(A low integrity subject will not write or modify high integrity data.)

** - Property*

If a subject s can have read access to object o , only if
 $I(o) \geq I(s)$.

(The high integrity subject will not read low integrity data.)

Restatement of the Biba rules:

- No write up.
- No read down.

Biba Example

Consider the following objects, subjects and integrity levels:

- File1 has an access class value of Administrator.
- File2 has an access class value of User.
- File3 has an access class value of Security Administrator.

- Subject1 has an access class value of Security Administrator.
- Subject2 has an access class value of User.

Where

- "Security Administrator" dominates "Administrator"
- "Administrator dominates User"

Under the Biba Model the following accesses are allowed:

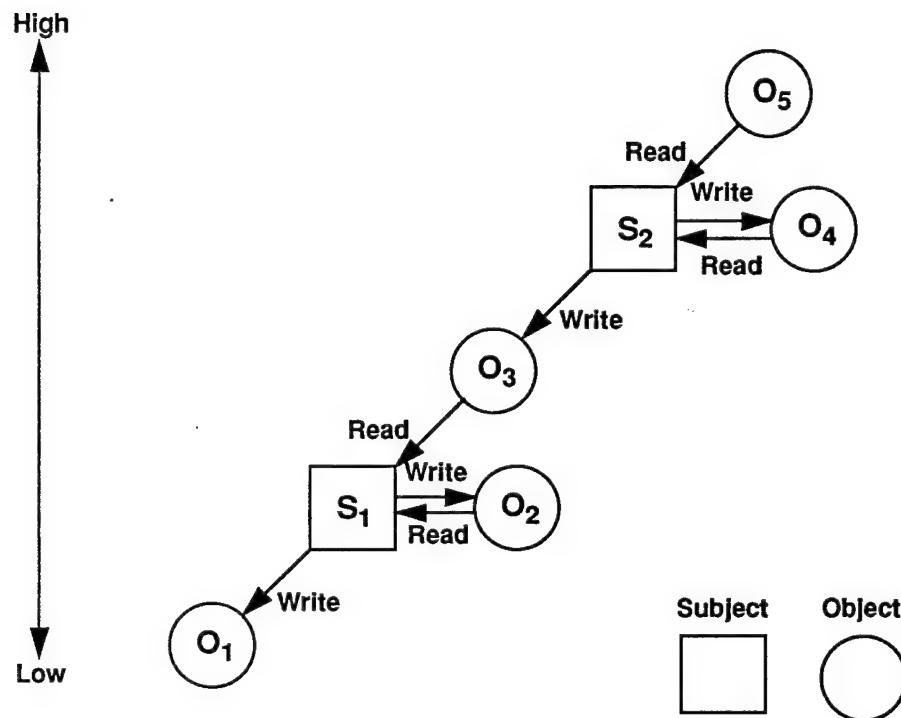
- Subject1 can read File3.
- Subject1 can write File1, File2 and File3.

- Subject2 can read File1, File2 and File3.
- Subject2 can write File2.

Biba Integrity Model

The Biba Model is often described in terms of secure information flows. The Figure below shows such a flow diagram. This is another way of saying that there is "no write up" and "no read down."

Integrity Flow (Biba)



As indicated by the diagram, a subject can only both read and write an object if the object has the same access class value as the subject.

Combined BLP and Biba Example

This example demonstrates how read and write accesses are restricted in systems that support the Bell and LaPadula secrecy model and the Biba integrity model.

- Each subject and each object has both a sensitivity label and an integrity label.
- Consider the following sensitivity levels TS, S, C.
- Consider the following integrity levels SecAdmin, Admin, User.
 - File1 TS-User
 - File2 TS-SecAdmin
 - File3 S-Admin
 - File4 C-SecAdmin
- Subject1 S-User
- Subject2 S-SecAdmin
- Subject3 TS-SecAdmin
- Subject4 TS-User
- Subject1 can read File3 and File4
- Subject1 can write File1
- Subject2, etc.
- The class TS-User is called system-high, because a subject at this class can read every object in the system. An object of this class can be written by every subject in the system.
- The class C-SecAdmin is called system-low, because it can be read by every subject in the system. A subject of this class can write every object in the system.

MAC Conclusions

Concluding statements about MAC.

- A MAC policy can prevent malicious software (e.g., Trojan Horses) from directly leaking information from high to low.
- Recall that we trust users to not give the store away, but we generally can't say the same thing for software.
- So we build systems that enforce a MAC policy on applications and we don't have to worry about the application software.
 - For example, a subject running at Secret cannot write any information at a level below Secret.

Note that a Trojan Horse can write information between objects at the same security level.

- For example a Trojan horse can read one Secret file and copy it to another Secret file.
- Is this a problem?
- No. Here's why.
- This scenario would require a bad guy (e.g., Ivan) to have a Secret clearance. (So you need personnel security too.)
- He brings in his killer Star Trek game (with an embedded Trojan Horse).
- Sue, a Secret user, plays the Star Trek game and the Trojan Horse copies her Secret files into Ivan's directory. But Ivan is already cleared for Secret information so the Trojan Horse does not get him any information he is not already cleared to see.
- In general, systems that support a MAC policy also support a DAC policy to provide a convenient separation of user's data.

There is still a potential problem with MAC systems.

- Covert Channels can still leak information from high to low in spite of a MAC policy.

Covert Channels

Covert channels are flows of information between access class levels counter to a MAC policy but which are allowed by an implementation.

- Covert channels are a means of leaking information from high to low, one bit at a time.
- If the rate of transmitting bits across the channel (the channel baud rate) is great, this threat is significant.
- Covert channels involve two programs, of which one must be a Trojan Horse. Covert channels are a little complicated to implement.
- However, if information being stored is very valuable, the covert channel threat is real.
- Covert channels come in two varieties. Storage channels and timing channels.
 - Covert storage channels exploit a resource common to both a high subject and a low subject.
 - Automated flow analysis tools can identify every storage channel in a formal specification of a system's interface.
- Covert timing channels exploit a mechanism where a high subject can affect the timing of low subject.
- No automatic means exist for identifying every existing timing channel at a system's interface.
- Timing channels are identified by a examination of the interface.

Covert Storage Channel Example

The classic example of a covert storage channel is the disk exhaustion channel.

- Ivan, (a low user) introduces a Trojan Horse program (e.g., Star Trek game) into the system and somehow gets a high user to execute it.
- When the high user plays the Star Trek game a sub-program is spawned and goes to sleep. The sub-program contains the Trojan Horse and wakes up and starts running at a time when activity on the system is low (e.g., at 0100).
- Ivan starts another program (a low program) that will wake up at 0105, (5 minutes later than the high program). This allows the high program time to initialize the channel.
- The high program finds a high file to copy (fileA).
- The high program initializes the channel by repeatedly creating files until the "disk full" exception is returned.
- The two programs will synchronize with each other by reading a system clock. The high program will signal bits on every even millisecond and the low program will receive bits on every odd millisecond.
- The high program starts reading the bits out of FileA. The following steps are repeatedly performed until the high program is through reading the file.
 - The high program does: (on even milliseconds)
 - If a bit is a 0, the high program deletes one file. (Creating room on the disk for a file to be created.)
 - If a bit is a 1, the high program does not delete a file. (So there is no room on the disk to create a file).
 - The low program does: (on odd milliseconds)
 - The low program always tries to create a file. If there is room on the disk, the create file call is successful. If the call is successful, the low program writes a 0 into a destination file.
 - If there is no room on the disk, the create file call will fail, with the "disk full" exception. If the call is unsuccessful, the low program writes a 1 into the destination file.

Covert Channels

Storage Channel Example Conclusions:

- The channel baud rate of the previous example is 1 bit every 2 milliseconds.
- This is 500 bits per second, which is 30,000 bits per minute.
- The timing scheme used in the example is very conservative. Much higher baud rates are generally attainable.
- One way to close the disk exhaustion channel is to partition the disk into volumes and allocate each volume to a different security level. For example, volume 0 is for TS files, volume 1 is for S files and volume 2 is for C files.
- Under this partitioning scheme, a C subject cannot tell if the TS volume is full or not. Recall that in the covert channel scenario, the C subject determined if the disk was full by attempting to create a file. Under the partitioning scheme C subjects create files on a separate volume than the TS subjects.

Covert Timing Channels

- Covert timing channels exploit a mechanism where a high subject can affect the timing of a low subject.
- A potential timing channel, which exists on single processor systems, uses the fact that both the high subject and the low subject use the same physical processor.
- To signal a 1, the high subject performs a lengthy operation (e.g., disk I/O) and signals a 0 by performing a short operation.
- When the high subject finishes its operation, the low subject is scheduled to run.
- When the low subject gets scheduled, it reads the system clock and determines how long the high subject operation took.

Multilevel Subjects

Multilevel subjects versus Single-Level subjects

- Up until this point, all subjects in the MAC discussions were single level subjects.
- That is, a subject could both read and write at only one level.
- For example, a Secret subject could only both read and write Secret level objects. (This is required to prevent malicious software from writing high data to low objects.)
- Situations exist (e.g., information downgrading) where a subject needs to be able to both read and write over a range.
- For example, downgrading information from Secret to Confidential would require a subject to read information in Secret objects and write it to Confidential objects.

Thus, MAC implementations must provide some means for multilevel subjects.

- Note that the vast majority of subjects will still be single-level subjects, because multilevel subjects are subject to the Trojan Horse problem.
- Thus, any code that is used in a multilevel subject **MUST BE TRUSTED**.
- That is, it must be examined to determine that it does not contain a Trojan Horse.
- Often multilevel subjects are call *trusted subjects*.
- Besides downgrading, there are other selected applications that require a multilevel subject.

Applications of Multilevel Subjects

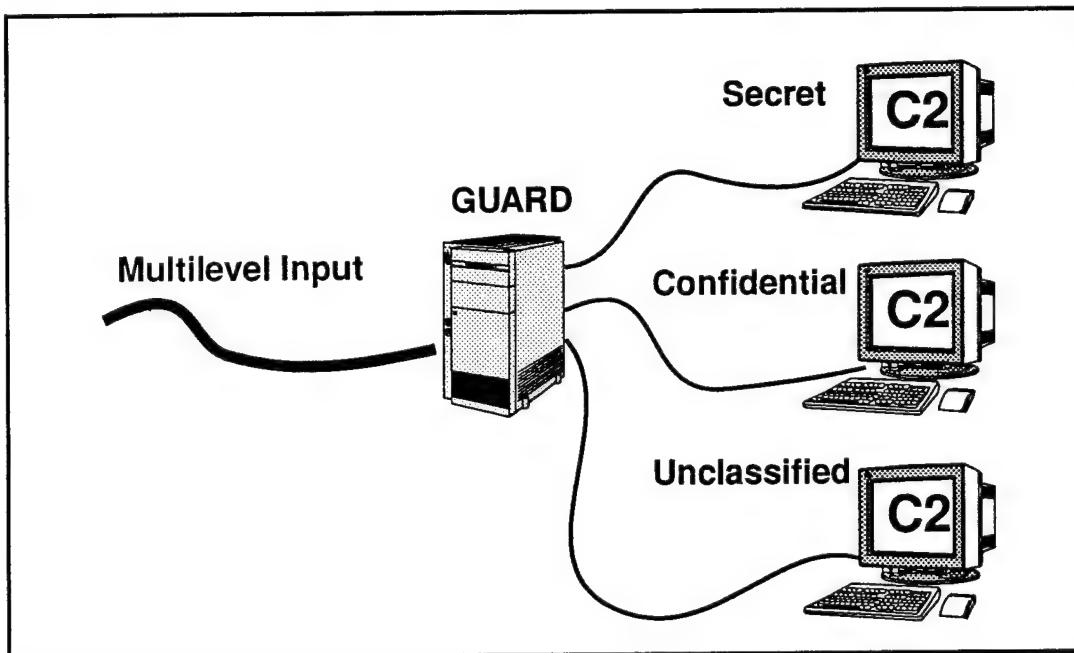
Down grading information

- Need human review - judgement
- Automated sanitization cannot assure that all sensitive information has been removed

Login to a Multilevel System

- users need to set session level
- user has HIGH and LOW range of security levels
- must set read and write class for session

Guards



Guard might consist of four processes:

- one multilevel process, P1 (multilevel),
- three single level processes P2 (S), P3 (C), P3 (U)
- P1 receives labeled information from multilevel input
- P1 inspects label and puts information into a labeled object
- P2, P3, or P4 send information from objects at its level to output system
- Guard is trusted to insure that labeled information is sent to the correct end system.

Supporting Policies

Supplement both mandatory and discretionary security policies

No matter how complex a policy may appear, if sufficient analysis is applied, it will be mandatory, discretionary, or supporting

Identification and Authentication

- associate subjects with users
- authenticate user to system and system to user (trusted path)

Audit - Accountability

Data Consistency Policy

- protects against damage resulting from user or software error
- protects against unauthorized modification or destruction of information
- E.g., Ages are non-negative, department credit card purchases must not be greater than \$200.00.

Accountability Policy

- authentication of individuals, thus permitting them to be accountable for the actions
- auditing of individual accesses and access attempts
 - deterrent to misuse
 - detection of security violations

Labeling Policy

- assignment of access labels to information entering and leaving the system
- assignment of access class authorizations to users

Aggregation Policy

- labeling of aggregates more sensitive than individual elements

Sanitization Policy

- release of derived information which is at a lower class than that from which it was derived.
- release of information from aggregates where the individual elements are at a lower class than the aggregate

Supporting Policies (continued)

Reclassification Policy

- classification changes raising or lowering the sensitivity of information

Applicability of Supporting Policies			
Supporting Policy	Mandatory	Discretionary	Dependency
identification and authentication	x	x	classification dependent for I&A on multilevel system
reclassification	x		classification dependent
labeling	x		classification dependent
sanitization	x		classification dependent
aggregation	x		classification dependent
consistency		x	classification independent
accountability	x	x	

This Page is intentionally blank

Section 4

Building Secure Systems I

*TCBs, Reference Monitors, Protection
Domains, Subjects and Objects.*

Assurance Versus Policy

Security Policy

- The previous two sections discussed several security policies and supporting policies for computer systems.
- These policies state rules that are enforced by a system's security features.

Assurance

- Assurance, within the context of computer security, is a measure of trust or confidence that a system's security policies are correctly enforced.

Note:

- Security Policies and Assurance are orthogonal. The number and type of policies enforced by a system says nothing about how well the policies are implemented. Assurance addresses the issue of how well a policy is implemented (with respect to correctness of the policy being enforced).
- The amount of effort required to analyze a system's ability to properly enforce its security policies, is dependent on the amount of software, firmware and hardware responsible for implementing the security features of the system.
 - A small amount of software can be analyzed with a reasonable amount of effort.
 - A large amount of software (e.g., an entire Unix operating system) **cannot** be fully understood and analyzed with **any** amount of effort. Large systems are beyond the scope of current analyzing tools and techniques.
- If we want to be able to analyze and fully understand the security features of a system, we either:
 - Build only small systems.
 - Build systems such that the security relevant code is small and separable from the non-security relevant code. Thus, only the small amount of security relevant code needs to be analyzed.

Trusted Computing Bases

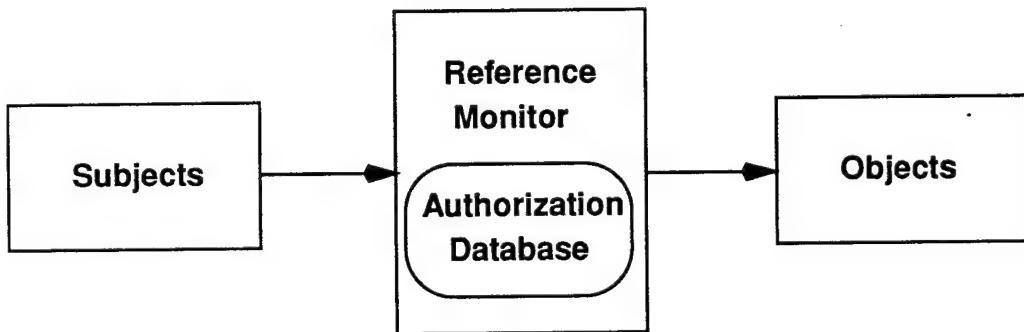
Trusted Computing Base Issues

- A Trusted Computing Base (TCB) is the totality of protection mechanisms within a computer system, including hardware, firmware and software. That is, the TCB contains the security relevant software, firmware and hardware.
- An imaginary boundary around the TCB is called the *security perimeter*.
- The TCB contains mechanisms for implementing the various security policies enforced by a system, (MAC, DAC, I & A, Audit, etc.).
- Of these policies the most crucial is MAC (recall that DAC is inherently flawed due to its susceptibility to malicious software).
- Special design and implementation requirements are needed for the portion of the TCB that implements MAC.
- These special design and implementation requirements lead to the *Reference Monitor Concept*.

Reference Monitor Concept

General Schema:

- The Reference Monitor is an abstraction that allows subjects to access objects.
- The Reference Monitor is interposed between all subjects and objects.
- The Reference Monitor makes reference to an authorization database.
- At an abstract level, the Reference Monitor supports two classes of functions:
 - Reference functions - for accessing information
 - Authorization functions - change authorization database



Reference Monitor

Implementation Requirements

Completeness

- The Reference Monitor must be invoked on every reference of a subject to an object.

Isolation

- The Reference Monitor and its database must be protected from unauthorized alteration.

Verifiability

- The Reference Monitor must be small, well-structured, simple and understandable so that it can be completely analyzed, tested and verified to perform its function properly.

Support Functions

- Reference Monitors often utilize supporting policies.

Identification and Authentication

- identify users to the system - who you are
- authenticate users to system - what you have, know, or are
- reliably identify trusted part of system to users

Audit

- record security relevant operations
- introduction of new objects into a domain
- deletion of objects
- create an **audit trail** composed of **audit records**
 - reference monitor may be source of only some of the audit trail information

Security Kernel

Security Kernel Facts

- A *Security Kernel* is an implementation of the reference monitor concept.
- It includes hardware, firmware and software.
- It demonstrates
 - Completeness
 - Isolation
 - Verifiability

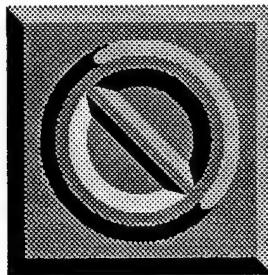
Conclusion

- A system that is built upon a Security Kernel:
 - Lends itself to a tractable analysis to determine *how well the system enforces specific security policies*.

Addressing Computer Misuse

- Don't need the power of a Security Kernel to address
 - user errors: best countered by user education
 - abuse of authority: countered by audit
 - direct probing: countered by sound management and audit
- Security Kernel particularly suited for addressing several categories of computer misuse:
 - probing with malicious software: countered by MAC policy
 - penetration: countered by high assurance systems
 - subversion: countered by high assurance systems

Security Kernel from an Existing Operating System?



NO!

Security Kernel Design using Existing Operating System

- Security functions may be diffused throughout system.
- Massive redesign required to
 - isolate security relevant functionality
 - insure modularity
 - insure use of hardware features in support of security kernel objectives

Determining Assurance

Aspects of Assurance

- Section 11 (Building Secures Systems II) covers several aspects of determining the level of assurance of a particular system.
- The remainder of this section covers:
 - **The use of Security Models to help establish assurance.**
 - **System architecture as it relates to assurance.**

Security Model issues:

- A Security Model is a precise and unambiguous statement of a systems security policy.
- A Security Model is an obvious representation of the security policy.
- A Security Model is simple and abstract, and therefore is easy to comprehend.
- An Informal Security Model may be written in formal mathematical notation or in a natural-language.
- A Formal Security Model is written in formal mathematical notation.

Security Models

Two ways a system may be insecure

- flaws in policy
- flaws in implementation

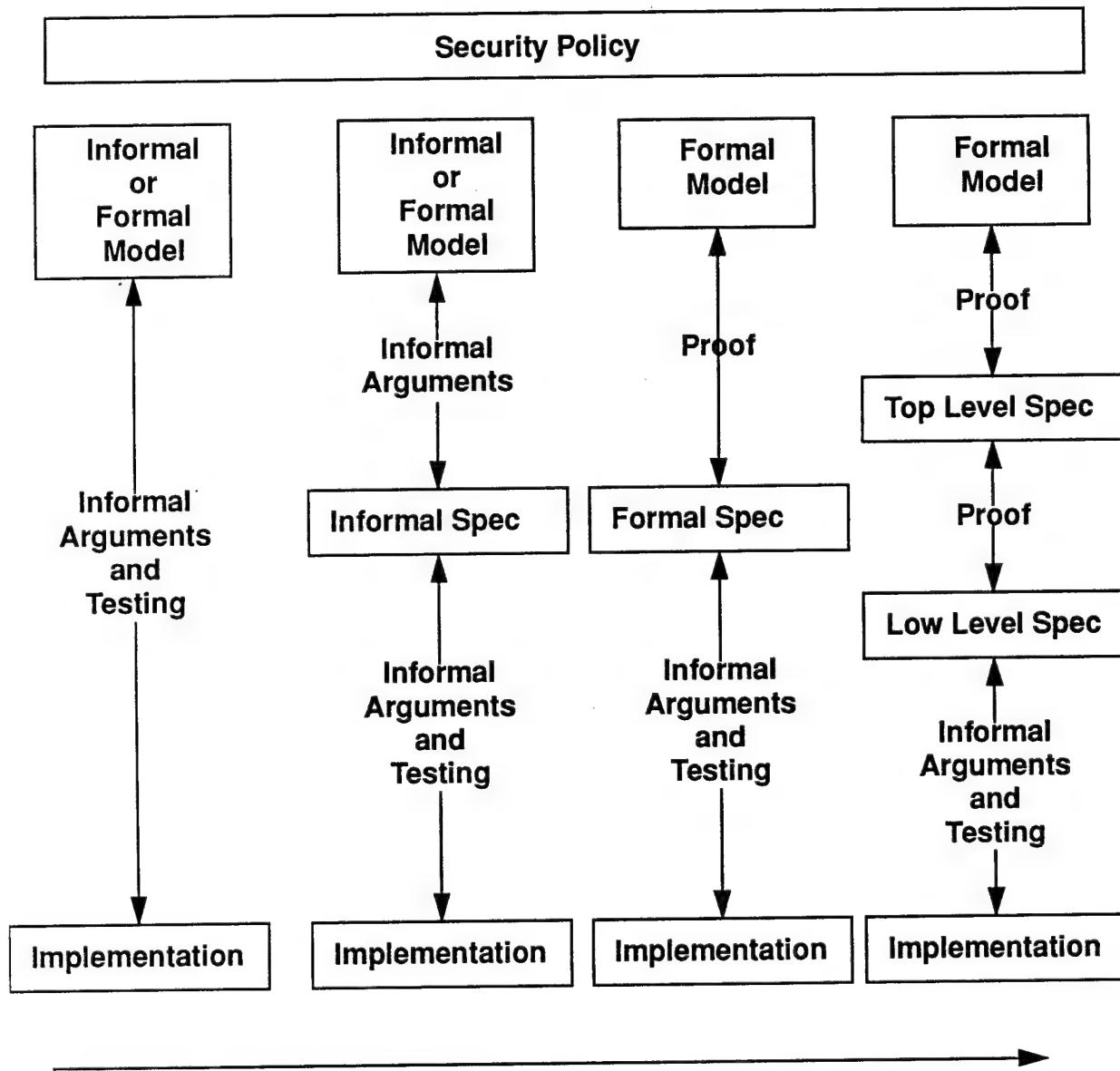
Reasons for using a Security Model

- Security Models can address both issues mentioned above.
- Both Informal and Formal Security Models can be used to establish that a Security Policy is not flawed.
 - We do not want to implement a system that enforces a flawed policy.
 - In the case of Formal Security Models, this proof is made by mathematically showing that the Model is consistent with its axioms.
- Both Informal and Formal Security Models can be used to establish that an implementation faithfully reflects the security policy.
 - An Informal Security Model can be mapped to an implementation or an Informal Specification which can help establish that an implementation faithfully reflects the enforcement of a security policy.
 - A Formal Security Model can be mapped mathematically to a Formal Specification which can help establish that an implementation faithfully reflects the enforcement of a security policy.

Formal Work Improves Verification

Objective:

Demonstrate that the implementation is faithful to the policy.



Increasing assurance that implementation is necessary and sufficient to enforce policy

General Characteristics of Security Models

Model constrains design to meet security requirements

- Doesn't constrain how that design might be implemented.
- Guides security relevant behavior of the mechanism.
- Mechanism expressed in a functional specification.

Do you always need a model?
No
Add-ons to an existing system are an example of weak security fixes where modeling would be useless.

Three Major Types of Models

State Machine Models

- State variables represent (security) state of machine.
- Transition functions describe changes to the variables.
- Access Matrix Models are state machine models.
 - Access matrix model shows how matrix changes using transition functions.
- Attribute model shows how security attributes of subjects and objects are compared.

Information Flow Models

- Control flow of information from one object to another.
- Useful for covert channel analysis.

Non-Interference Models

- Subjects operate in different domains and are prevented from affecting each other in ways that would violate the security policy.
- Still active research topic.

State Machine Modeling Steps

Preliminary Steps

- Define relevant security state variables.
- Define what it means to be in a secure state.
 - (e.g., all Unclassified subjects don't have read access to any Confidential objects.)
 - **This is called the *invariant*.**
- Define the state transition functions.
- Select an initial state for the system.

Proof Steps That Establish the Model is Consistent With its Axioms

- Prove that the initial state is secure.
- Prove that each individual function maintains a secure state (i.e., that they take a secure state to another secure state).

Induction is the basis of the proof.

- Since the initial state is secure and
- since all state transition functions maintain a secure state (take a secure state to another secure state)
- all combinations of transition functions can only result in a secure state.

Conclusion:

If a system starts in a secure state and all transition functions take a secure state to another secure state, the system will always be in a secure state.

Example Access Control Matrix

Abstract Representation of an Authorization Database:

- A table in which:
 - Each row represents a single user (subject).
 - Each column represents an object.
- Access matrix can be represented as a list of triplets to avoid sparse matrix. $\langle \text{subject}, \text{object}, \text{rights} \rangle$

		Objects							
Subjects		Bibliog	Temp	Test.tnp	Help.txt	C_Compiler	Linker	Sys_Clock	Printer
User_A	RW	RW	RW	R	X	X	R	W	
User_B	R	-	-	R	X	X	R	W	
User_S	RW	-	R	R	X	X	R	W	
User_T	-	-	-	R	X	X	R	W	
SysMGR	-	-	-	RW	X	X	RW	-	
User_Svcs	-	-	-	-	-	X	X	R	W

Access Control Matrix

In the example above the rights are

R - read

W - write

X - execute

Simple State Machine (SSM) Example

More details? See Gasser, *Building a Secure System*

Policy

A person may read a document only if the person's clearance is greater than or equal to the classification of the document.

Policy to Model Translation

Policy terms	Model terms
people/paper world	computer world
person	subject
document	object
clearance	access class
classification	access class

Property 1: A subject may read an object only if the access class of the subject is greater than or equal to the access class of the object.

Property 2: A subject may write to an object only if the access class of the object is greater than or equal to the access class of the subject.

- Note that Property 1 is the Bell and LaPadula simple security property and Property 2 is the BLP *-property.

SSM: Define State Variables

Notation:

- \emptyset null set
- $\{ \}$ set notation
- \cup union
- \forall for all
- \in is an element of
- $'x$ the value of x in the state after a transition (i.e., in the next state)

S = set of current subjects

O = set of current objects

$sclass(s)$ = access class of subject s

$oclass(o)$ = access class of object o

$A(s,o)$ = set of access modes equal to one of:
 $\{r\}$ subject s can read object o
 $\{w\}$ subject s can write object o
 $\{rw\}$ subject s can read and write object o
 \emptyset neither read nor write access

$contents(o)$ = contents of object o

$subj$ = active subject

state of system at any time is

$\{S, O, sclass, oclass, A, contents, subj\}$

Define Secure State

Invariant: the system is secure if and only if, $\forall s \in S, o \in O,$

if $r \in A(s,o)$, then $sclass(s) \geq oclass(o)$,

if $w \in A(s,o)$, then $oclass(o) \geq sclass(s)$,

SSM: Define Transition Functions

Function	Effect
create_object (o, c)	create object o at class c
set_access (s, o, modes)	set access modes for subject s to object o
write_object (o, d)	write data d into contents of o
create/change_object(o,c)	set class of o to c and create
copy_object(from, to)	copy contents (from) to contents (to)
append_data(o, d)	add data d to contents of o

Functions are defined mathematically and are atomic operations.

create_object (o, c)

if $o \notin O$
then ' $O = O \cup \{o\}$ ' and
' $oclass(o) = c$ ' and
 $\forall s \in S, 'A(s, o) = \emptyset'$

set_access(s, o, modes)

if $s \in S$ and $o \in O$
and if { [$r \in modes$ and $sclass(s) \geq oclass(o)$] or $r \notin modes$ }
and
{ [$w \in modes$ and $oclass(o) \geq sclass(s)$] or $w \notin modes$ }
then ' $A(s, o) = modes$ '

Notes

- $=$ means mathematical equality not programming assignment.
- The order of statements not important.
- Transition functions must be atomic.
- If something isn't described in the function then nothing happens to it, everything that changes in the state must be described in the function.

SSM: Proof of Consistency

Prove Each Transition Function

Invariant and Function imply 'Invariant'

Example: create_object proof

$\forall s \in S, o \in O, \text{ if } r \in A(s, o), \text{ then } sclass(s) \geq oclass(o),$

$\text{ if } w \in A(s, o), \text{ then } oclass(o) \geq sclass(s),$

and

$\text{ if } o \notin O$

$\text{ then } 'O = O \cup \{o\} \text{ and } 'oclass(o) = c \text{ and } \forall s \in S, 'A(s, o) = \emptyset$

implies

$\forall s \in 'S, o \in 'O, \text{ if } r \in 'A(s, o), \text{ then } 'sclass(s) \geq 'oclass(o),$

$\text{ if } w \in 'A(s, o), \text{ then } 'oclass(o) \geq 'sclass(s),$

Note how the create_object function needs to force nulls in the column of the access matrix for the new object. Needed for the function to be secure.

Define and Prove Secure Initial State

$\{S_0, O_0, sclass_0, oclass_0, A_0, contents_0, subj_0\}$

Simple Initial State

$S_0 = \emptyset \text{ and } O_0 = \emptyset$

Another Initial State

$\forall s \in S_0, o \in O_0, sclass_0(s) = c_0, oclass_0(o) = c_0$

$A_0(s, o) = \{r, w\}$

Conclusion

After each transition function is proved secure (i.e., each transition function takes a secure state to a secure state) and an initial state is proved secure, the model proof is complete. This means that the model is consistent with its axioms and that the security policy is not flawed.

SSM: Constraints

Must insure that transitions from state to state are secure

Add constraints - these address values in two consecutive states

- maintain secure relationship between "old" and "new" values
- restrict subjects from invoking certain operations under certain conditions
- control transitions that modify information

Example

change/create _ object (o, c)
'oclass(o) = c ; and
if $o \notin O$ then ' $O = O \cup \{o\}$; and
 $\forall s \in S, 'A(s, o) = \emptyset$

Problem: this function permits the access class of an object to be changed. Information could be downgraded.

Solution: add a new property

Property 3: the access class of an object cannot decrease

We are dealing with a particular type of transition, so add a constraint constraint: $\forall o \in O, 'oclass(o) \geq oclass(o)$

Important Models

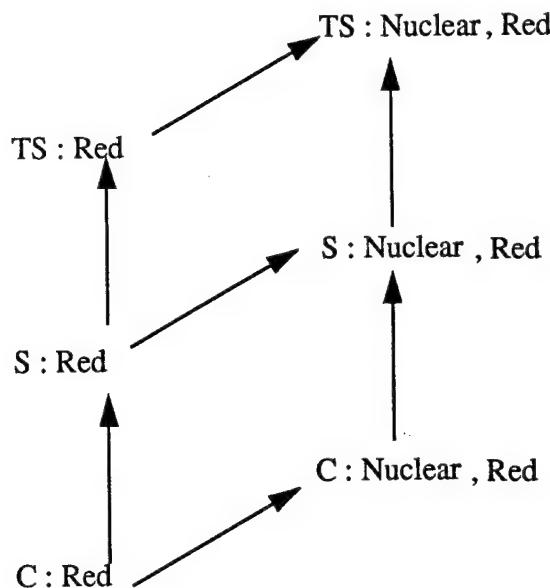
Computer Security literature often makes reference to the following Security Models:

- Lattice Model
 - A lattice model is a generalized model where the elements form a mathematical structure called a lattice.
 - The figure on the following page shows a lattice representation of the legal information flows within the context of military style labels.
 - Many other models satisfy lattice model properties.
 - Models that exhibit lattice properties lend themselves to mathematical analysis.
- Bell and Lapadula Model
 - It can be expressed in terms of a lattice model.
 - It is a confidentiality model.
 - It was the first mathematical model of a multilevel secure computer system,
- Biba Model
 - It can be expressed in terms of a lattice model.
 - It is an integrity model.
 - It is the dual of the BLP model.
- Graham-Denning Model
 - An information flow model.
- Harrison-Ruzzo-Ullman Model
 - An information flow model.
 - A theoretically important model, which facilitates proofs regarding the decidability of subjects gaining rights to objects.
- Clark-Wilson Model
 - It is a commercial model that is transaction oriented.

Example Lattice

Example lattice showing legal information flows in a system that has:

- Secrecy levels of "TS", "S" and "C"
- The compartments "Nuclear" and "Red"



- Notice how information flows from label "A" to label "B" only if label "B" dominates label "A".
- Least Upper Bound (LUB)
 - The LUB of a set of lattice elements (or levels and compartments) is defined to be the "least dominant" element that dominates all elements of the set.
- Greatest Lower Bound (GLB)
 - The GLB of a set of lattice elements (or levels and compartments) is defined to be the "greatest dominant" element that is dominated by each element of the set.
- Examples:
 - The LUB of "S : Red" and "C : Nuclear, Red" is "S : Nuclear, Red"
 - The GLB of "TS : Nuclear, Red" and "S : Red" is "S : Red"
- System-High is the upper bound of all security classes in a system.
- System-Low is the lower bound of all security classes in a system.

System architecture issues:

System architecture considerations directly address the isolation requirement of Security Kernels. Important specific issues are:

- The use of hardware to support domains of execution.
- The use of hardware to support distinct storage objects.
- The use of Layering, Modularity and Data-Hiding.
 - Layering is the structuring of software into distinct loop-free layers (i.e., layers only call down).
 - This allows software to be analyzed in smaller chunks (one layer at a time) since the correctness of a lower layer is not affected by an upper layer.
 - Modularity is the structuring of software into small understandable single purpose chunks.
 - Data-Hiding is the structuring of data such that it can only be manipulated through a simple high-level well defined module interface.
- The use of the principle of "Least Privilege".

Principle of least Privilege

A subject should have access to the fewest objects needed for the subject to work successfully.

(Information is limited by a need-to-know!)

Example:

The system backup program may be allowed to bypass read restrictions on files, but it need not have the ability to modify files. The restore program might be allowed to write files but not read them.

OS Protection of Memory

An OS may offer protection of memory (system objects) at any of several levels:

- No protection (unrestricted sharing between processes and subjects)
- Isolation (no sharing between processes subjects)
- Restricted sharing between processes and subjects
 - Share via access limitation
 - Share by capabilities

Types of protection:

- Physical Separation
- Temporal Separation
- Cryptographic Separation
- Logical Separation

Hardware can provide support for protection

- registers
- privilege levels
- privileged instructions

Simple (Early) Protection Schemes

Fences

Fixed Fence

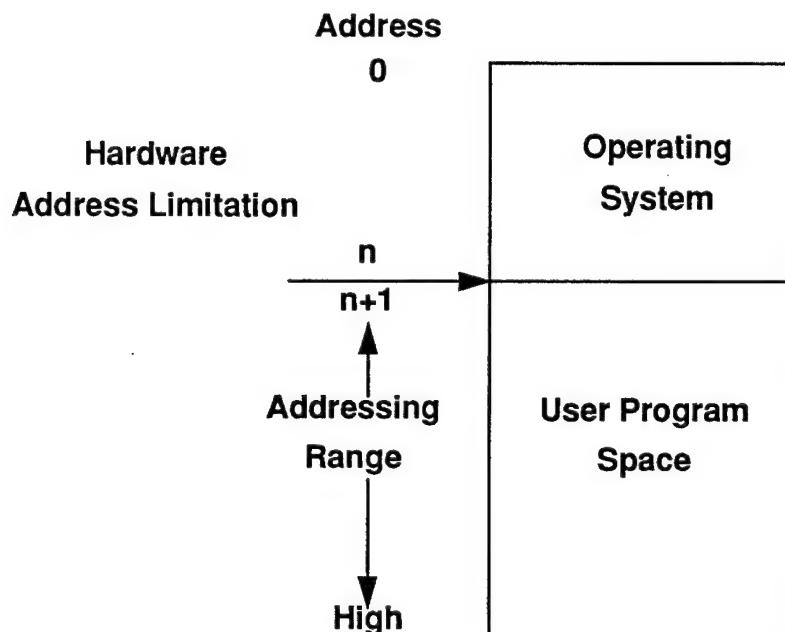
- A method to confine users to one side of a boundary.
- Predefined memory address (fixed).
- Operating system on one side.
- User program on the other side.

Relocation:

- The process of taking a program written as if it began at address 0 and changing all addresses to reflect the actual address at which the program is located in memory.

Fence register:

- Contains address of end of OS.
- Provides means of code relocation.
- Only protects operating system.
- Does not protect one user from another

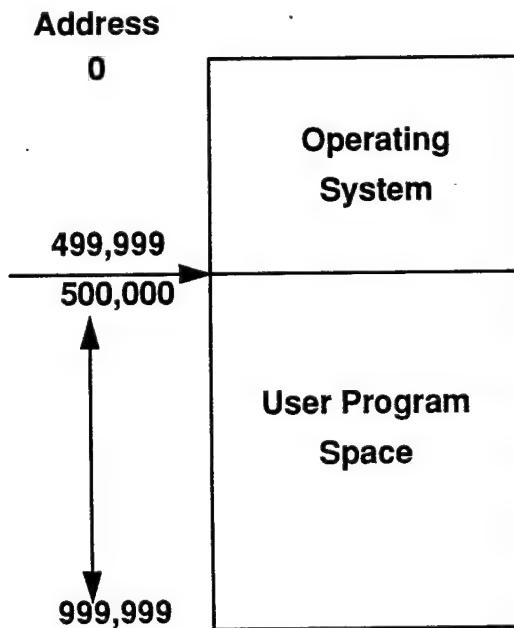


Fixed Fence

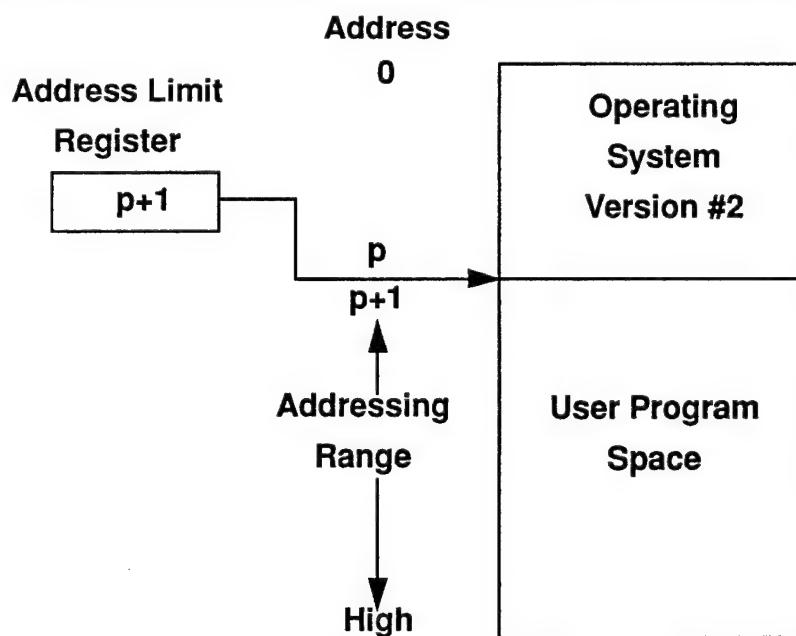
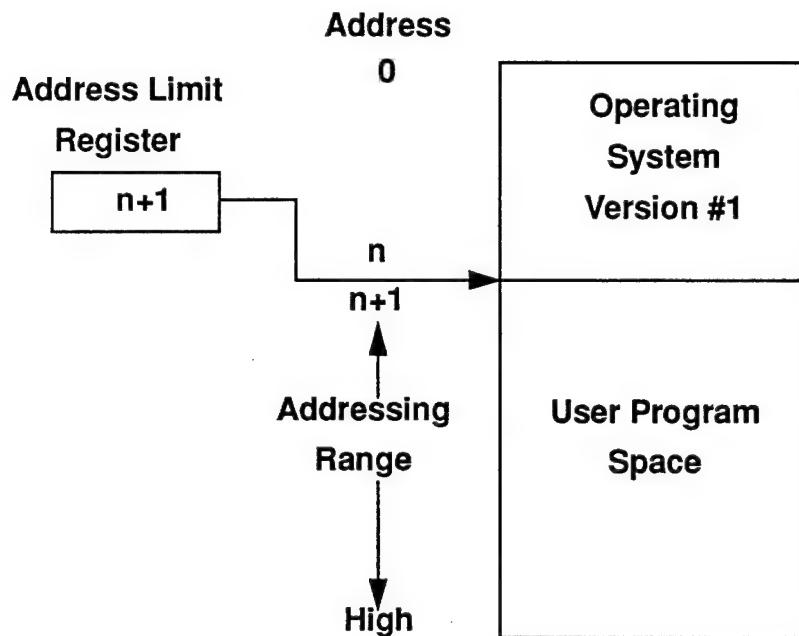
Process / Subject Distinction

The difference between a Process and a Subject:

- A Process is a thread of execution.
- A Subject is a Process executing in a domain.
 - A domain is an address space (i.e., the totality of memory locations addressable by a process).
- In the situation shown below there are two domains for a process:
 - The OS domain (memory location 0 to 999,999) and
 - the User Program domain (500,000 to 999,999),
- When process PROC1 is executing a User Program, it is restricted to the memory range 500,000 to 999,999. When the User Program needs an OS service (e.g., writing to a device), PROC1 makes an OS call and PROC1 starts executing in the OS. While PROC1 is executing in the OS, it is restricted to the memory range 0 to 999,999. When PROC1 finishes the requested OS service, PROC1 returns to executing within the User Program.
- The OS subject of PROC1 is when PROC1 is executing in the OS.
- The User subject of PROC1 is when PROC1 is executing a User Program.



Protection of Memory and Addressing



Variable Fence Register

Protection of Memory and Addressing

Base/Bounds Registers

Base register:

- All addresses are offset from the base register.
- A variable fence register generally called a base register.

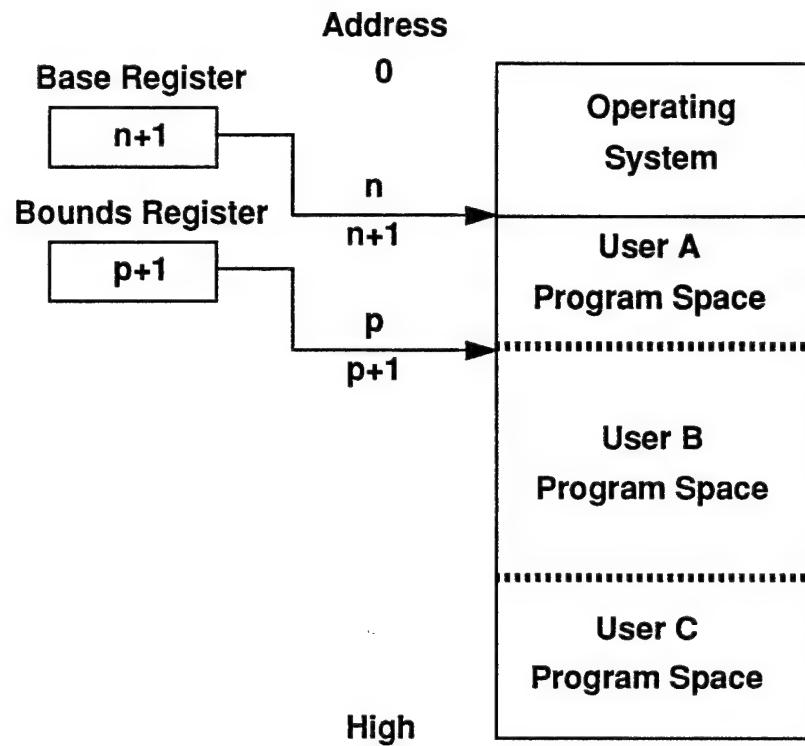
Bounds Register:

- Provides upper address limit.
- When used with a base register user program is confined.
- Provides one user's program protection from another.
- Does not protect user from himself/herself.
 - Can be achieved through additional registers

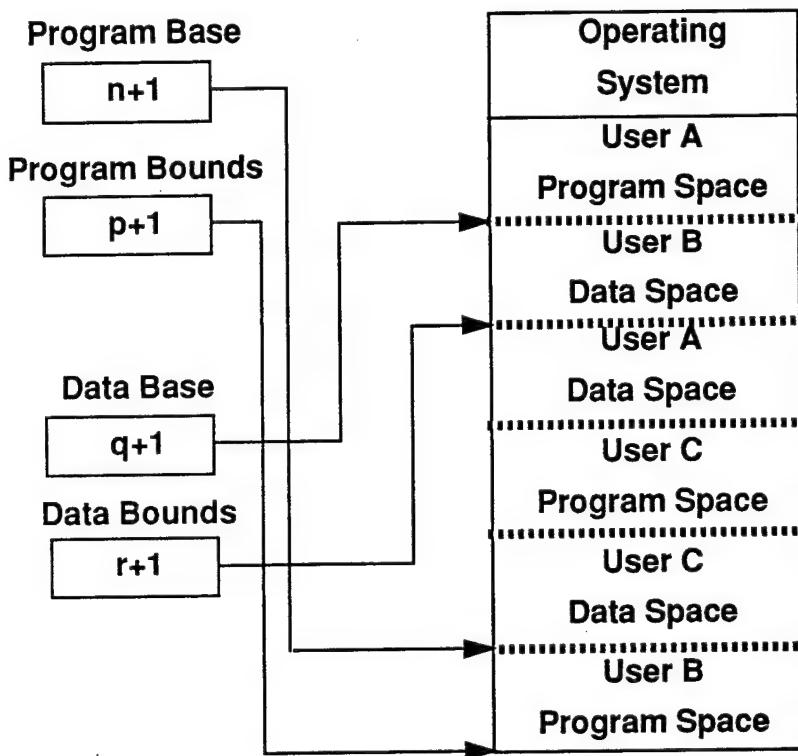
Virtual Machine Supervisory Program:

- Generally the only process which can change the contents of these registers.
- Maintains a protected table of all register value pairs (one for each virtual processor).

Protection of Memory and Addressing



Protection of Memory and Addressing



Two Pairs of Base/Bounds Registers

- The Program registers specify the range of memory addresses allowed for code.
- The Data registers specify the range of memory addresses allowed for data references.
- A general scheme would also include a memory area for each process stack.
- Given that each process will require a set of three register values, this scheme gets a little complicated.
- A scheme that simplifies the management of many different memory regions (code, data and stack for Process A, code, data and stack for Process B, etc.) is the memory segmentation scheme.

Segmentation of Memory

Segmentation

- Segmentation is the notion of dividing memory into separate pieces, called segments.
- Each segment has a unique name.
- Individual bytes of memory are addressed as a pair <segment name, offset>
- The O/S maintains a table mapping the logical addresses to the physical addresses.
- This scheme has the same effect as an unbounded number of base/ bounds register pairs

A form of information hiding:

- The OS can move any segment to any location.
- A segment can be removed from main memory.
- Every address reference passes through the OS.
 - A process which does not have a segment name in its table is denied access to that segment.
 - Handled by combination of hardware and software.

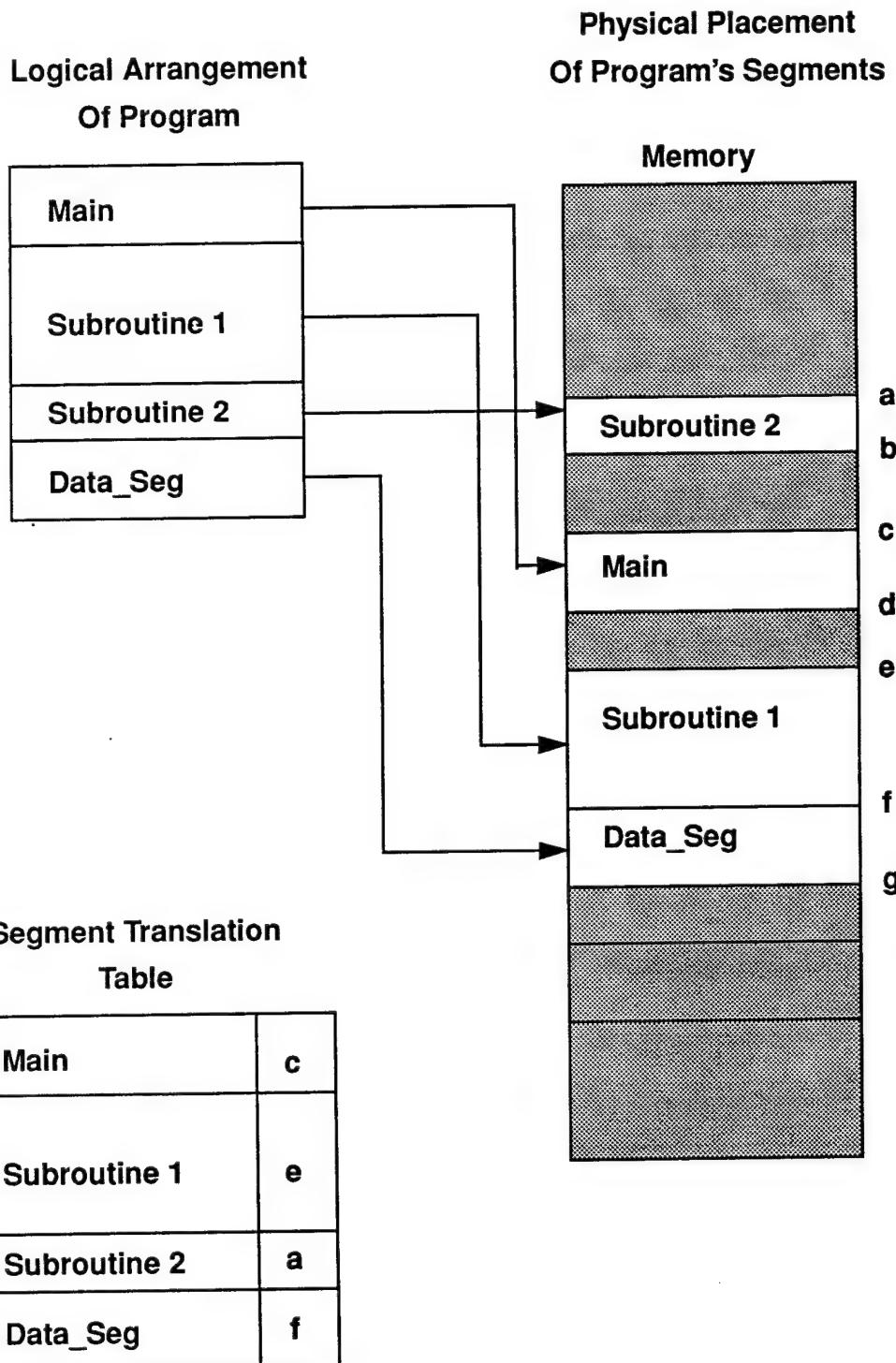
Benefits of segmentation:

- Each segment can be assigned a different level of protection (e.g., access class label or value).
- Each request for a segment can be checked for appropriate access (perfect for Reference Monitor / Security Kernel implementations).
- Two or more users can share access to a segment, but with different access rights.
- It is impossible for a user to generate an address or gain access to an unpermitted segment.

Inherent problems:

- Segment names are inconvenient to encode.
- Segmentation leads to fragmentation of main memory.
- If swapping is used then additional memory management techniques must be employed (e.g. LRU).

Protection of Memory and Addressing



Logical and Physical Representation of Segments

Segmentation Continued

Use of segmentation:

- Each process has a table of segments that it can access (Intel uses the terminology "Descriptor Table").
- This table specifies the address space of the process.

Paged Segmentation

- Paged memory schemes are convenient since they automatically manage the task of swapping in and swapping out pages of memory as needed by programs.
- Break each segment into equal sized pages.

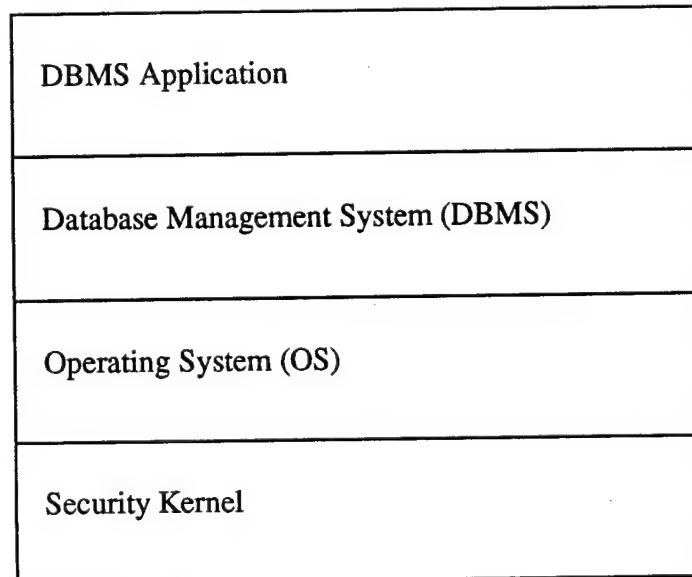
Advantages of hardware that supports the segmentation of memory:

- Supports the isolation of processes by providing a simple means for specifying process address spaces.
- Supports labelling of objects (perfect for Reference Monitor / Security Kernel implementations).
- The next few slides will show how segmentation can support the implementation of execution domains.

Execution Domains

Motivation:

- The Process / Subject discussion a few pages back assumed the existence of two execution domains, the OS domain and the User Program domain.
- In general more than two execution domains are desired to support the principle of least privilege.
- The figure below shows an architecture that uses four distinct domains of execution for each process:
 - It can indirectly affect data in the lower domains by making calls to that domain.
 - For example, the Application can ask the DBMS to modify a database table, but it cannot directly affect the table without using the DBMS.



Execution Domains Continued

Implementation Details of Execution Domains:

Execution Domains can either be implemented in software or hardware.

- Intel x86 (8086, 80286, 80386, 80486 and Pentium) chips support four hardware execution domains.
- Intel uses the terminology "Hardware Privilege Level".
- Multics hardware supported 32 domains.
- Execution domains are sometime referred to as *rings*.
- The term *hardware rings* implies hardware enforced domains and *software rings* implies software enforced domains.
- An important aspect of a ring implementation is the mechanism that allows outer (less privileged) subjects to call into inner (more privileged) subjects.
- This mechanism is called a *call gate*.

Call Gate Issues:

- Call gates (or gates) limit the way the less privileged subjects invoke the services (procedures and functions) of the more privileged subjects.
- Gates specify exactly which entry points of a domain may be called from above. For example, the Security Kernel ring in the previous figure may contain 323 functions and procedures, but the Security Kernel gate only allows the upper subjects to call 27 of them.
- Gates prevent upper subjects from jumping into the middle of procedures and functions in the lower domain.
- Gates can also be used to limit which subjects may call into a domain. For example, in the previous figure, the Security Kernel gate may only allow calls from the OS subject. That is, the DBMS subject cannot call directly into the Security Kernel.
- Gates are also used to validate pointers that are passed into a procedure of a more privileged ring. The gate ensures that all pointer parameters don't point to any addresses that are part of the more privileged domain.
- Intel x86 chips provide robust gate mechanisms for supporting the four privilege levels.

Why Care about Hardware Privilege Levels?

Hardware Versus Software Implementation

- Hardware is far more efficient than software.
- High Assurance Products Use Hardware Mechanisms.

Hardware Platforms of High-Assurance Trusted Products

Trusted Product	Target Rating	Base Architecture
Boeing MLS LAN	A1	80x86 Multiprocessor
Gemini Trusted Network Processor	A1	80x86 PC or Multiprocessor
Wang XTS 300	B3	80x86
Verdix VSLAN	B2	80x86 Custom Board
TIS Trusted Xenix	B2	80x86 PC

Table from: Sibert et al, *The Intel 80x86 Processor Architecture: Pitfalls for Secure Systems*

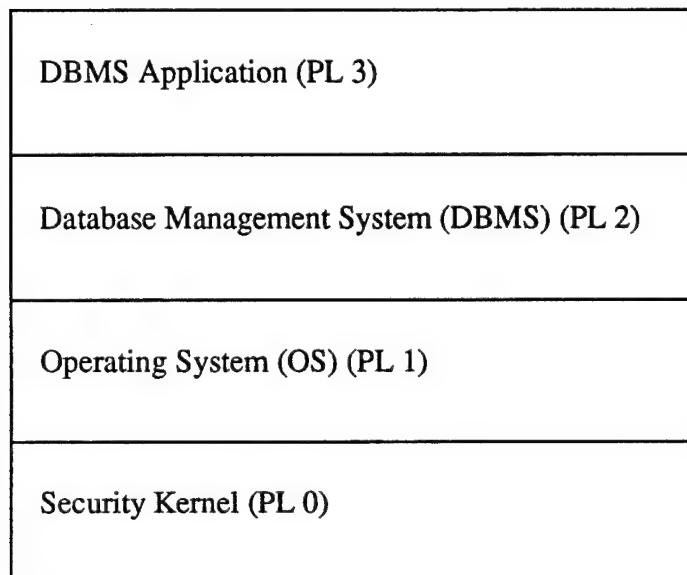
Ring Issues

Software Rings

- With software rings, the Security Kernel creates the ring abstraction, (i.e., it enforces the ring policy).

General Ring Issues

- A process may run in any one of several rings at any one time, moving from ring to ring during execution.
- A process running in a given ring is protected from other processes running in the same ring (process isolation).
 - Recall, a process running in a given ring is called a subject.
- Intel Privilege Levels are denoted as PL 0 through PL 3.
 - See the figure below.
- Ring mechanisms enforce a ring policy.
- A common ring policy is where a subject running in ring i can access all data and functions in ring j, if $i \leq j$.
 - This is the policy enforced by the Intel hardware Privilege Level mechanisms.



Ring Brackets

Ring Bracket Motivation

- Ring brackets are often part of ring enforcement mechanisms.
- The use of ring brackets allow for a ring policy that is more robust than the policy enforced by Intel hardware Privilege Levels.
 - See the example below.
- A set of ring bracket values are associated with each segment of memory.
- The following example demonstrates a ring bracket policy and encoding.

Example

- Three ring bracket values R_1 , R_2 and R_3 are associated with each segment.
 - 0- R_1 is the write bracket
 - 0- R_2 is the read bracket
 - 0- R_3 is the execute bracket

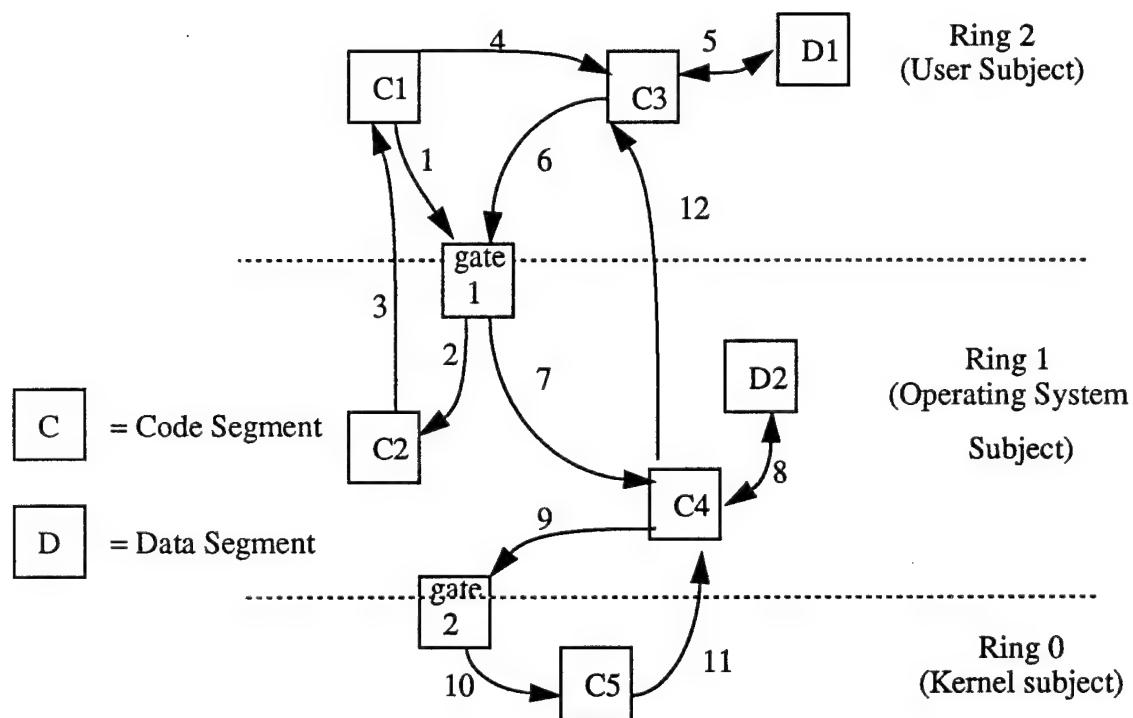
Ring Bracket values:	R_1	R_2	R_3
----------------------	-------	-------	-------

A segment with ring brackets of (4,5,7) is writable from rings 0 through 4, readable from rings 0 through 5, and executable from rings 0 through 7. (it is assumed that $R_1 \leq R_2 \leq R_3$)

Example of Subjects, Processes and Gates

Single Process, Multi-subject Scenario

- In the figure below, execution starts in the upper left-hand code segment, C1.
- Subsequent processing goes as follows:
- C1 calls a procedure (via arrows 1 and 2) in C2. It needs to go through gate 1.
- The procedure in C2 finishes and execution returns (via arrow 3) to the calling point in C1.
- C1 (via arrow 4) calls a procedure in C3.
- C3 reads and writes data (via arrow 5) from/to D1.
- C3 (via arrows 6 and 7) calls through gate 1 to a procedure in C4.
- C4 reads and writes data (via arrow 8) from/to D2.
- C4 (via arrows 9 and 10) calls through gate 2 to a procedure in C5.
- The procedure in C5 finishes and execution returns (via arrow 11) to the calling point in C4.
- Similarly the procedure in C4 returns to C3 (via arrow 12).



Design of Secure Operating Systems Overview

Qualities of Secure Systems

1. Security policy - well defined and enforced by system.
2. Identification - every subject must be uniquely identified.
3. Marking - objects labeled for comparison when access requested.
4. Accountability - must maintain complete and secure records.
5. Assurance - must contain mechanisms which enforce security and must be able to measure their effectiveness.
6. Continuous protection - mechanisms must be protected themselves.

Basic Considerations

- Security must be considered in every aspect of the design of operating systems.
- It is difficult to add on security features.

Principles of Design

1. Least privilege - fewest possible privileges for user.
2. Economy of mechanism - protection system should be small, simple and straight forward.
3. Open design - mechanism should be open to scrutiny.
4. Complete mediation - check every access.
5. Permission based - default permission should be denial of access.
6. Separation of privilege - one permission should not give away the entire system.
7. Least common mechanism - avoid shared objects.
8. Easy to use.

This page is intentionally blank.

Section 5

Malicious Software and Intrusion Detection

Malicious Software

Greedy programs

- Background task assumes greater priority.
- May not be malicious in nature.
- Infinite loops (are you guilty?)
 - Most systems use time-outs.
 - I/O time not usually checked.

Trapdoor

A secret, undocumented entry point into a module.

- Inserted sometime during code development.
 - Most often debugging hooks.
 - May permit direct change of variables.
 - Produce unwanted side effects.
- Poor error checking.
 - Unacceptable input not caught.
- Most instances are not malicious in nature.
 - Even if not malicious others may utilize it.

Trojan Horse

Performs a hidden function in addition to its stated function.

- Generally distributed as object code along with documentation of overt use.
- Micro users particularly susceptible.
- Can be introduced by binary manipulation (DEBUG).
- Instructions may be scattered with jumps.
- Instructions may be encrypted.

New Threats

- Malicious remote executables.
 - Downloaded Java
 - Agentware
- Really variations on known problems.

Malicious Software

Viruses

Self replicating and infectious program.

- May be relatively harmless or disastrous.
- More prevalent on PCs
 - Do not compile their own source code like mainframes.
 - Swapping of programs.
 - Easier to get infected (opportunity is there).
 - Ignorance.
- Viruses can be categorized by:
 - How do they infect others?
Overwriting virus?
Non-overwriting virus?
 - Where do they live?
Boot infector?
System infector?
Application infector?(specific or generic)
- Viral hiding techniques:
 - Self encrypting to avoid detection of the signature.
 - Polymorphic to present a different signature every time.
 - File compressing to be able to hide in files without increasing the size of the host file.

Worm

A program that can run independently and can propagate a fully working version of itself to other machines!

- Does not require the host program to be run to activate it (as is the case for a virus).
- Not all are malicious
 - file compression routines
 - automatic back-up routines

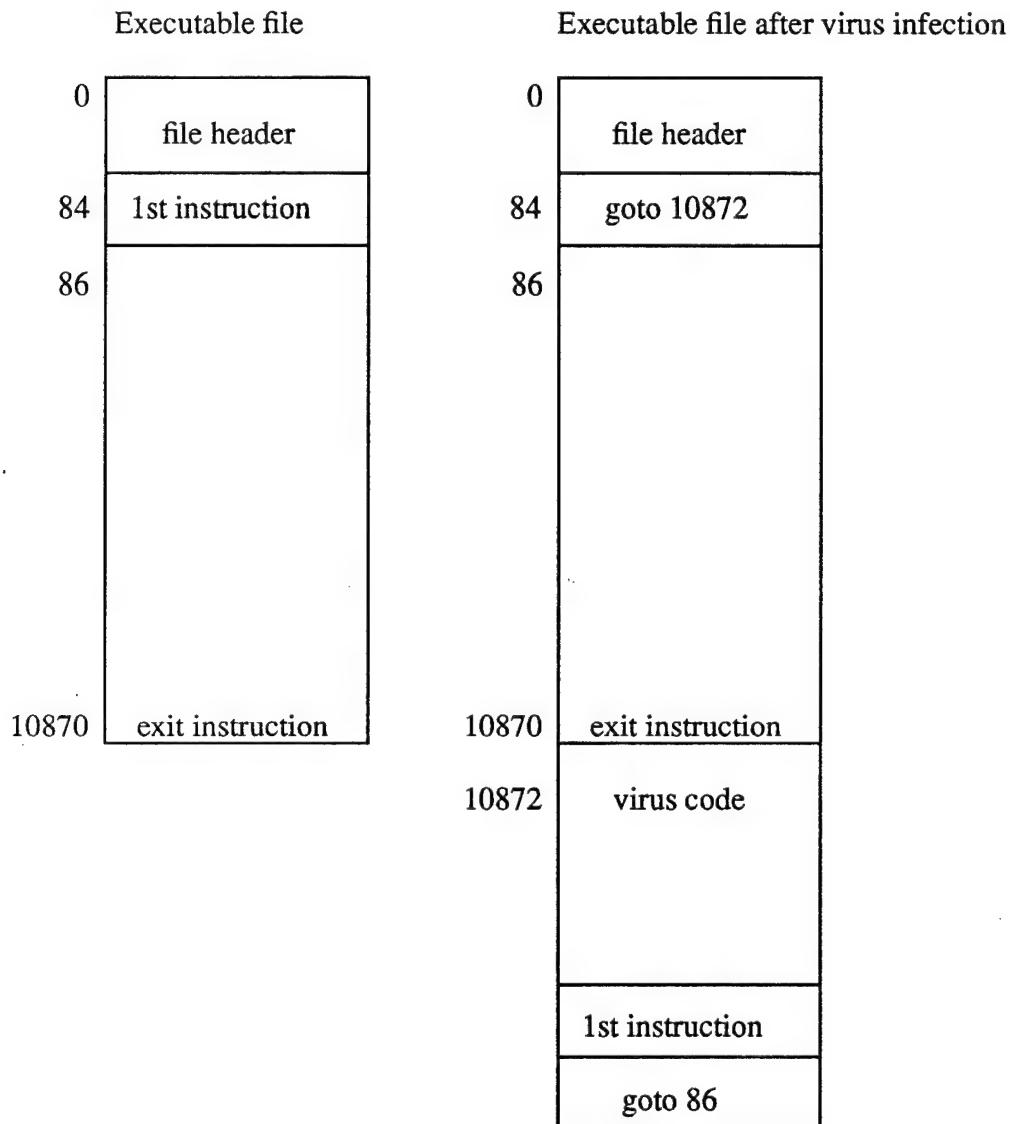
Viruses

Historical overview

- 1949 - John von Neumann publishes “Theory and Organization of Complicated Automata” a report which dealt with the subject of self reproducing code.
- 1960’s - Bell Lab programmers H. Douglas McIlroy, Victor Vysotsky, and Robert Morris, create an after hours recreation game called “Core Wars” which experiments with code designed to reproduce and gain control of the computer’s core memory.
- “Core Wars” concept becomes a popular pass time at several other industrial and academic research centers but remains a closely guarded secret.
- 1970’s -Several futuristic novels, notably: “Shockwave Rider” by Thomas Brunner and “The adolescence of P-1” by Thomas J. Ryan feature worms and intelligent, information-seeking viruses.
- 1984 - Disclosure of recipe for “Core Wars” virus revealed to general public.
- 1986 - University of Delaware comes under attack by both the Brain virus and the Scores virus.
- 1988 - Princeton University requests assistance to combat *nVir* virus which was attacking their systems. Three hours later Stanford University reported a similar incident. One week later another *nVir* virus attack took place at Oulu University in Finland.
- March 1992 - Michelangelo virus is broadly publicized and overall the general public becomes more aware of the destructive potential of computer viruses.
- MS Word macro virus
 - Really executed
 - CERT advisory

Viruses

General Virus Infection Scheme



How could this threat be limited or prevented?

- By isolating incoming code so that it cannot modify system code or important applications.
- By using file access control mechanisms (DAC or MAC).
- Antiviral software.

Viral Infection Rates

Viral Infection Models vs. Reality

- A number of researchers have developed models, based upon biological models, in an attempt to describe the infection rate of viral infections.
- These models, which have been around since the late 1980's, suggest that every computer system in the world should be infected with some form of virus today. We know this is not true, so these models fail to account for other factors which reduce the overall effect of viral infections.
- There are several reasons why these models do not accurately model the real world. Two obvious reasons stand out from all the rest:
 1. The models do not accurately model the effect of anti-viral precautions.
 2. The models do not take into account the effect of wide scale virus detection/removal upon discovery of new viruses.

The Best Strategy in the Anti-Viral War

Education of the user community and rigid anti-viral efforts!

Viral Infection Sources (Source: Dataquest, 1992)

- 43% Disk from home
- 25% Don't know / refuse to say
- 7% Electronic Bulletin Board (EBB)
- 6% Sales demo disk
- 6% Repair or service disk
- 3% Shrink-wrapped application
- 2% Download other than EBB
- 2% Intercompany disk
- 1% Came with PC

The Virus Threat

Preventing a Virus Infection:

- Use only commercial software acquired from reliable, well-established vendors.
- Test all new software on an isolated system.
- Make backup bootable installation and recovery disks.
 - Keep disks write-protected during reboot.
- Use virus detectors.
 - Can be configured to run periodically or when new files are imported.

Intrusion Detection

Definition:

- The ability to detect security lapses, ideally while they occur.

Techniques:

- Intrusion detection software builds usage patterns of the normal system and triggers an alarm any time the usage is abnormal.

Issues:

- Related to audit reduction.
- Helpful against insiders.
- The field is still young.

This page is intentionally blank.

This page is intentionally blank.

This page is intentionally blank.

Section 6

*Accreditation,
Certification,
Disaster Planning,
and
Risk Analysis*

Accreditation

Definition - Accreditation

Formal declaration by a designated approving authority (DAA) that an AIS is approved to operate in a particular security mode and environment using a prescribed set of safeguards.

Official Management authorization to operate a system.

Grants approval to operate

- In a particular security mode
- Prescribed countermeasures
 - physical
 - administrative
 - emissions
 - computer security (COMPUSEC)
 - communications security (COMSEC)
 - personnel
- Against a defined threat
 - with stated vulnerabilities, countermeasures
- Within given operational concept and environment
- Stated system interconnections
- Acceptable level of risk
 - accrediting authority assumes risk responsibility
- Specified Time Period

Accreditation Results

- Accreditation affixes security responsibility with DAA
- Shows that due care has been taken for security in accordance with policies
- Given parameters of accreditation, system will protect against
 - compromise, modification, destruction
- Accreditation is in effect
 - upon formal acceptance
 - for specified period
- Accreditation required prior to processing live data (waiver required otherwise)
 - classified
 - sensitive but unclassified
- Interim approval possible

Accreditation Issues

DOD Directive 5200.28

5200.28 Requires Accreditation, March 1988

Classes of Information

- Classified
- Sensitive but unclassified
- Unclassified

Applicable systems

- Stand alone
- Communications
- Network - digital, hybrid, analog
- Peripheral devices and software
- Process control computers
- Embedded computer systems
- Communications switching computers
- Personal Computers
- Intelligent terminals
- Word processors
- Office automation systems
- Applications and operating system software
- Firmware
- Etc.

Accreditation Issues

Safeguard Information

Classified and Sensitive But Unclassified

- Access only by authorized individuals
- Used only for intended purpose
- Retain content integrity
- Properly marked

Unclassified

- Safeguard against
 - Tampering
 - Loss
 - Destruction
- Helps prevent fraud, waste, abuse

Safeguarding Techniques

- Administrative
- Procedural
- Physical/environmental
- Personnel
- Communications security
- Computer security
- Emanations Security

Accreditation Supported by:

- Certification Report
- Risk Analysis

Designated Approving Authority

Definition - DAA

The official with the authority to formally assume responsibility for operating a system at an acceptable level.

A.K.A.:Accrediting Official
Accrediting Authority

The more sensitive the system, the more senior the DAA

DAA Authorities

- evaluate overall mission requirements of system
- provide definitive direction relative to risk in system security posture

System Responsibilities

- One system - One DAA
- Multiple systems - One DAA
- One System - Multiple DAA
 - Careful agreements must be established

DAA Responsibilities

Each DAA shall

- Review and approve security safeguards
 - issue accreditation statements
 - based on safeguard acceptability
- Ensure that safeguards are implemented and maintained
- Identify security deficiencies
 - take action
- Name an Information System Security Officer (ISSO)
 - Adequate training
 - Directive recommends that ISSO not report to operational elements of AIS over which security requirements of the 5200.28 directive must be enforced
- Require AIS security education and training be in place
- Establish data ownership
 - accountability
 - access rights
 - special handling requirements

DAA Identification

Factors

- type of information:
 - SCI Sensitive Compartmented Information
 - SIOP-ESI Single Integrated Operational Plan-Extremely Sensitive
 - collateral
- office of primary responsibility (owner) of system
- interconnections with separately accredited systems
- Sometimes the owner of the data defines the accreditor
- Sometimes the accreditor is defined by the owner of the system

DAA Maps to Security Policy

- Single DAA
 - Agency/Service Policy
 - DoD Policy
- Multiple DAAs
 - Multiple Agencies
 - Cannot know details of each other's policies
 - system may be subject to requirements of all organizations
 - clearly define policies
 - document through MOA

New Network Connections?

System DAA must consider risks

Network DAA must consider security of system requesting connection

- Must comply with network security requirements prior to connection

Network Accreditation Issues

Interfacing or Networking

AISs managed by different DAAs

MOA required and includes

- description and classification of data
- clearance levels of users
- designation of DAA who shall resolve conflicts among DAAs
- safeguards to be implemented before interfacing AISs

Networking Responsibilities:

- Multilevel Communications Network
 - e.g. World Wide Military Command and Control System Intercomputer Network
 - One DAA responsible for overall security
- Safeguards agreed to, implemented and accredited prior to hook up
- DAA in charge may remove non-compliant AIS from network
- May have subnets
 - use cryptography from NSA or DIA to define boundaries
- Highest accreditation division required based on security requirements

Multiple DAA and Security Policy Issues

National Security Policy

Agency and service Security Policies

- Total of 34 Policies

Example: AUTODIN/DMS

- Mode of System Operation
 - Multilevel
- Policy: DCAC 370-195-3
 - Defence Communications Agency (DCA), March 1987, *DCS AUTODIN Category III Operational Acceptance Test*, DCA Circular (DCAC) 370-195-3
- Multiple DAAs
 - DISA/DA
 - DIA/DS-SIM
 - JCS/(DJS/J6)
 - NSA/Office of operational security

DAAs and policies required for SCI data.

Info type	Policies	Navy DAA	Marine Corps DAA	Army DAA	Air Force DAA
SCI	DCID 1/16 DCID 1/19 DIAM 50-3 DIAM 50-4	DIA/DS-SIM DNI	DIA/DS-SIM Director, Intelligence Div. (CMC Code INT)	DIA/DS-SIM MACOM Commander, Heads of DA Staff Agencies (dedicated mode); HQDA (DAMI-CIC-AS)(other modes)	DIA/DS-SIM HQ USAF/INS, AFIS/IND

Certification

Definition

The comprehensive analysis of the technical and nontechnical security features and other safeguards of a system to establish the extent to which a particular system meets a set of specified security requirements

- Supports accreditation process
- Targets specific environment

Certification includes

- Risk Analysis
- Security Testing
- Evaluations
- etc.

Certification Considers

- Mission security requirements
- Operational environment

Certification Personnel

- Technically competent to assess system
- Produce documentation provided to DAA

Part of System Lifecycle

- Track security state of system
 - changes to system
 - changes to environment

Environmental Factors Considered

- Mode of Operation
- Users
- Applications
- Data Sensitivity
- System Configuration
- System/Site Location
- Interconnections

Test system security attributes against threats in the intended environment

Factors Affecting Certification

Mission Security Requirements

Compare

- Fixed format, high integrity, on-time messages
- Office automation mail

System Complexity

Compare

- Stand alone PC
- LAN with workstations, file server, WAN gateway

Risk Environment

- data sensitivity
- user clearances
- mission criticality
- external interfaces

Previous C&A evidence

- Previously evaluated products
- Products or subsystems that have already been accredited

Role Of Risk Management

Definition

The process concerned with the identification, measurement, control and minimization of security risk in information systems

- Encompasses entire system lifecycle
- Has direct impact on system certification

Identify Areas for Safeguards

- Disclosure
- Modification
- Denial of Service (DOS)
- Unauthorized use

Apply Countermeasures to Risks

- Eliminate
- Reduce

May Include

- Risk Analysis
- Cost-Benefit analysis
- Countermeasure selection
- Security test and evaluation
- Countermeasure implementation
- Penetration testing
- System review

Iteratively Applied

System design -->

 Countermeasure specification

System implementation -->

 verify effectiveness of countermeasures

 identify additional countermeasures

Operational system -->

 verify effectiveness of countermeasures

 identify additional countermeasure

Management commitment to risk management must be defined as early as possible in program lifecycle

Recertification and Reaccreditation

Part of stated Policy

DoD policy

System shall be reaccredited every three years

Director of Central Intelligence

System shall be reaccredited every five years

System Changes

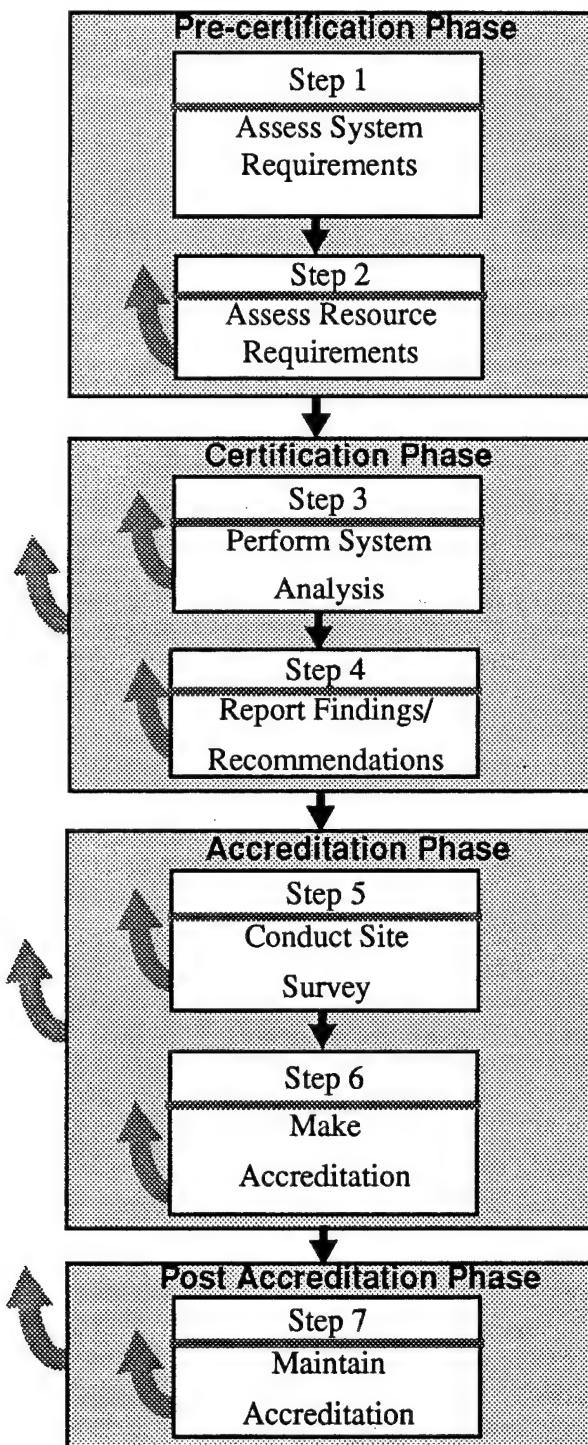
- Change in criticality and/or data sensitivity
- Change in security policy
- Change in threat or system risk
- Change in security mode of operation
- Changes in operating system or software providing security features
- Changes in hardware providing security features
- Breach of security, integrity, or other unusual situations which might invalidate accreditation
- Physical changes
- Configuration changes (e.g., new connections)
- In networks,
 - changes to subscribing systems
 - addition of other accredited systems
- Results of audit or external analysis

Separately Accredited Networks

AUTODIN/DMS

Unclassified (but Sensitive) IP Routing Network (NIPRNET)
Secure IP Routing Network (SIPRNET)
Joint Worldwide Intelligence Communications System (JWICS)
Defense Information Systems Network (DISN)
Integrated Tactical Strategic Data Network (ITSDN)
Critical Intelligence Communications (CRITCOMM) System
Special Intelligence Communication (SPINTCOM) Network
STU III (Secure Telephone Unit-III)
Red Switch

Certification and Accreditation Process



Certification and Accreditation Process

Step 1:

- Involves gathering and developing relevant documentation (e.g., policy implementation guidance, security regulations/manuals, previous certification reports, product evaluation reports, COTS manuals, design documentation, design modification, and security related waivers). Aspects to be considered during this step include:
 - Mission criticality
 - Functional requirements
 - System security boundary
 - Security policies
 - System components and their characteristics
 - External interfaces and connection requirements
 - Security mode of operation or overall risk index
 - System and data ownership
 - Threat information
 - Identification of the Designated Approving Authority (DAA)

Step 2:

- Since security should have been considered with system conception, planning for C&A is a natural extension of system security planning. That is, the schedule and resources required to complete the C&A process are identified. Aspects to be considered during this step include:
 - Reusability of previous evidence
 - Life-cycle phase
 - System milestones (time constraints)

Certification and Accreditation Process

Step 3:

- The security aspects of the system as a whole (i.e., how well security is employed throughout the system) is analyzed. C&A activities during this step include determining whether system security measures adequately satisfy applicable requirements.

Step 4:

- This step involves documenting /coordinating the results and recommendations of previous phases to prepare the certification package which is used as supporting documentation in the accreditation package. The types of documentation generally included as part of the certification package include:
 - System need/mission overview
 - Security policy
 - Security Plan
 - System architectural description and configuration
 - Reports of evaluated products from a recognized government evaluation.
 - Statements from other agencies indicating that personnel, COMSEC, or other security requirements have been met.
 - Risk analysis report
 - Test plans, Test procedures and test results from security tests conducted
 - Analytic results
 - Configuration Management Plan
 - Previous C&A information
 - Contingency plan
 - Memoranda of understanding (MOAs)

Step 5:

- This step is optional and involves the DAA or his/her representative conducting a site survey to assure that security requirements meet the requirements for the system.

Certification and Accreditation Process

Step 6:

- DAA makes the accreditation decision based upon factors such as global threats, system need/criticality, certification results and recommendations, residual risks, the availability or cost of alternative countermeasures and other factors that may reach beyond pure security considerations (e.g. political consequences). The DAA may decide:
 - Full accreditation
 - Accreditation for operation outside the originally intended environment (e.g. mission needs, crisis situation).
 - Interim accreditation with steps identified for accomplishment prior to full accreditation.
 - Disapproval.

Step 7:

- Accreditation maintenance throughout the life-cycle by ensuring that the system continues to operate within the stated parameters of the accreditation.

Protection from Natural Disasters

Natural Disasters

Impossible to Prevent but possible to reduce the damage

Flooding

- Natural
 - rain
 - tides
 - overflows
- Artificial
 - broken water pipes
 - sprinklers
- Rising Water
 - time is on our side
 - hardware is replaceable
 - real concern is data and programs
 - most centers do not label media by priority groups
 - locate computer center above ground level

or

 - place building on elevated ground
- Falling Water
 - flooding on upper floors seeps down
 - sprinkler system goes off
 - large plastic bags
 - covers or bags near all terminals

Protection from Natural Disasters

Fire

- Time is critical
- Emphasizes need for quick, orderly shut down procedures
- Smothering systems used vice water
- No windows/fire-resistant access doors

Power Loss

- Uninterruptible power supply

Power Drops/Spikes/Surges

- Surge suppressors
- Line conditioners
- Unplug computer
- Disconnect phone line

Heat

- Only solution is to shut down system
- Greatest problem is unreliability of performance
- Adequate ventilation/air conditioning
- Inspect air system regularly

Recovery Protection

Types of Backup

- Complete backup takes time
 - everything on the system is copied
- Revolving backup
 - each time a backup is done, the oldest is replaced
- Selective (partial) faster
 - only those files which have changed are copied
- Optimum is a proper combination of complete and selective
- Must be able to reconstruct from last backup to failure point
- Complete record of transactions since last backup

Offsite Backup

- Must store media and paper trail
- Backups right next to machine - NEVER

Cold Site

- A shell facility
- Own or rent
- Usually resume operations within a week

Hot Site

- Ready to run facility
- Company may own facility or subscribe to service
- May or may not be staffed

The Key to Successful Recovery

Complete and Timely Backups!

Disaster Planning and Recovery

Components of a Disaster Recovery Plan

- EDP Disaster Recovery Plan Report
 - assumptions and considerations
 - recovery requirements
 - descriptions of all resources reviewed, highlighting the critical resources
 - strategies considered and recommended strategies
 - detailed recovery procedures
 - emergency plan and backup plan
 - staffing and responsibilities
 - maintenance and testing procedures
- Recommendations for Actions
 - actions required to be taken by management to put the plan in place and test it regularly
- EDP Disaster Recovery Procedures
 - detailed assignments and locations for actions to be taken at the time of an emergency until the backup operation is running
- Recovery and Restoration Procedures
 - procedures to return to the original site or another one selected
- Documentation and Related Information
 - appendix which may not be included with all copies of the plan

Disaster Planning and Recovery

Phases of Disaster Recovery Study

I. Definition Phase

1. Decide on disaster recovery objectives.
2. Appoint planning coordinator and planning team.
3. Develop initial set of assumptions.
4. Decide on types of disasters to consider.
5. Tentatively select a key disaster scenario.

II. Functional Requirements Phase

1. Assemble all organizational procedures and standards relative to emergencies.
2. Assemble all documentation relative to the inventory of resources, including hardware, communications, software, forms, facility descriptions, etc.
3. Make an evaluation of what systems are mandatory, necessary, or desirable.
4. Analyze the applications and facilities against the recovery objectives.
5. Decide on long term strategy or short term high-impact plan.
6. Assess the operational requirements of the critical resources and applications.
7. Agree on the assumptions and definitions.
8. Tentatively determine what is to be included in the plan.
9. Set priorities and acceptable time-frames for recovery.

Disaster Planning and Recovery

III. Design and Development Phase

1. Decide on the requirements for the critical resources and applications.
2. Evaluate alternative recovery strategies.
3. Select one or more recovery strategies.
4. Perform a cost/benefit analysis for the management report.
5. Perform a full risk analysis.
6. Decide on the organization for the Disaster Recovery Teams.
7. Plan the management of resources during a disaster event.
8. Identify potential vendors and price services.
9. Select the final design and prepare detailed recovery procedures.
10. Produce the plan report with recommendations.

IV. Implementation Phase

1. Acquire any hardware, software, communications lines, etc. that are needed.
2. Negotiate and sign contracts with vendors.
3. Get agreement on final detailed procedures.
4. Train personnel.
5. Prepare sites.
6. Develop test and monitoring plans.
7. Develop maintenance plan.

V. Testing and Monitoring Phase

1. Set up test plan with internal review/audit.
2. Schedule tests for small sections of the plan at a time.
3. Make arrangements to use facilities external to your organization.
4. Attempt to run backup systems.
5. Correct errors in the plan.
6. Repeat a variety of tests periodically.

VI. Maintenance Phase

1. Develop a system to update names, responsibilities and telephone numbers.
2. See that system for backup libraries is working smoothly.
3. Standardize documentation and procedures.

Disaster Planning and Recovery

Content of An MIS Contingency Plan

I. Contingency Plan for Major Disasters

- A. Detection and Reaction
 - 1. Identifying the problem; notifying the authorities
 - a. Emergency Services
 - b. Environment
 - c. Physical security
 - 2. Reducing your exposure
 - a. Air-conditioner failure
 - b. Fire alarm procedure
 - c. Electrical failure procedures
 - d. Flood and water damage
 - 3. Evacuation of the facility
 - 4. Advising the Emergency Management Team of the situation
 - 5. Creating a flow chart of the detection and response
- B. Initiation of the Backup-site
 - 1. Emergency Management Team notifies other teams
 - 2. Establish Control Center
 - 3. Begin Disaster Recovery Team operations and Disaster Recovery logs
 - 4. Timed events
 - a. 1 to 6 hours after being notified
 - b. 6 to 12 hours after being notified
 - c. 12 to 24 hours after being notified
 - d. 24 hours after being notified
- C. Establishment of Full Recovery at Backup Site
 - 1. All planned software, hardware and resources in place at backup site, and the applications tested
 - 2. Communications network and other equipment in fully operational
 - 3. Disaster Recovery Team checklists
- D. Restoration of Facilities and Operations at the Original and/or Alternative Site

Disaster Planning and Recovery

II. Disaster Recovery Teams

- A. MIS Organizational Chart
- B. Descriptions and Responsibilities
 - 1. Disaster Planning Coordinator
 - 2. Emergency Management Team
 - 3. Operations Team
 - a. Computer operations
 - b. Facility preparation
 - c. Replacement hardware
 - d. Cold-site preparation
 - e. Computer support equipment
 - f. supplies
 - 4. Data Entry and Control Team
 - a. Data input
 - b. Data control
 - 5. Special Projects Team
 - a. Transportation to/from backup facilities
 - b. Training
 - c. Administrative services
 - 6. Technical Support Team
 - a. Systems software
 - b. Communications network
 - 7. Data Administration Team: Database Restoration and Integrity
 - 8. Systems and Programming Team
 - a. Application systems restoration and recovery
 - b. Application programs
 - 9. Insurance Department Team: Insurance and Salvage
 - 10. Internal Audit Team: Verification of the Integrity of Restoration
- C. Team Preplanning and On-Going Functional Responsibility
 - 1. Disaster Planning Coordinator
 - 2. Emergency Management Team
 - 3. Operations Team
 - 4. Data Entry and Control Team
 - 5. Special Projects Team

Disaster Planning and Recovery

6. Technical Support Team
7. Database Team
8. Systems and Programming Team
9. Insurance Department Team
10. Internal Audit Department Team

III. Data Center Requirements

- A. Computer Room and Tape Library Layout
- B. Power Requirements, Cable Diagrams, and Plug Connectors
- C. Air-Conditioning, Fire Protection, and Security
- D. Computer Equipment and Vendor Location and Serial Number
 1. Computer room
 2. Data entry
 3. Other areas: Programming/systems/technical services/offices
- E. Teleprocessing: Configuration Information
 1. Line flow chart drawing
 2. Communications controller
 3. Satellite
- F. Terminal Configuration Charts
 1. Local terminal configuration
 2. Remote terminal configuration

IV. Suppliers

- A. New and Used Hardware Suppliers
- B. Software Suppliers
- C. Communications Suppliers
- D. Special Equipment Suppliers
- E. Office-Support Equipment Suppliers
- F. Computer Custom-Forms Suppliers

V. Prioritize All Applications

- A. Rate all Systems with Their Priorities
- B. Assign Responsibility for all Applications
- C. Designate Systems Requiring Detailed Recovery Plans

Disaster Planning and Recovery

VI Media Protection

- A. Protection and Retention of Vital Records
- B. Protecting the Database
 - 1. Database backups
 - 2. Updates
 - 3. Database definitions
 - 4. Software modification source code
- C. Standard Backup Procedures
- B. MIS Staff
- C. Service and Maintenance Personnel
- D. Outside Company Personnel
 - 1. Hardware
 - 2. Communications
 - 3. Miscellaneous
- E. Access Control
- F. Secured Forms-Room Access
- G. Vault Access
- H. Non-office Hours
- I. Security Duties: Guards
- J. Office Security.

X. Software Security

- A. Sign-On Passwords
- B. Maintaining Application Programs
- C. Password Maintenance

XI. Backup Facilities

- A. Subscribing to a Backup Facility
- B. Facility Layout
- C. Hardware and Software
- D. Communications
- E. Supplies
- F. Testing
 - 1. Initial testing
 - 2. Restoring your files and libraries
 - 3. Testing critical applications

Disaster Planning and Recovery

4. Testing communications
5. Mock Disasters
6. Testing program compilations

XII. Reciprocal Agreements

XIII. Insurance Protection

- A. Data Processing Property Protection Coverage
- B. Insurance on Computer Hardware
- C. Insurance on Other Data Processing and Office Equipment
- D. Business-Interruption Insurance

XIV. Policing the Plan

XV. Maintaining the Contingency Plan

- A. Disaster Planning Coordinator's Responsibility
- B. Team Captain's Responsibility

Risk Analysis

Definitions

Vulnerability

A weakness in system procedures, system design, implementation, internal controls, etc., that could be exploited to violate system security policy.

Threat Agent

- A method used to exploit a vulnerability in a system, operation, or facility.
- Categories of Threat Agents include:
 - information gatherers (e.g. spies)
 - terrorists
 - organized crime
 - malicious criminals
 - pranksters
 - insiders
 - outsiders with access
 - “Mother Nature”

Risk

- The probability that a particular threat will exploit a particular vulnerability of the system.

Safeguards

- The protective measures and controls that are prescribed to meet the security requirements specified for a system.
 - Safeguards may include:
 - hardware and software security features
 - operating procedures
 - accountability procedures
 - access and distribution controls
 - personnel security
 - physical structures, areas and devices

Risk Analysis

Risk Analysis

- The process of identifying security risks, determining their magnitude, and identifying areas needing safeguards.
- Part of risk management.
- Risk analysis is a process.
- A risk assessment is the result of the risk analysis process.

Risk Management

- The total process of identifying, controlling, and eliminating or minimizing uncertain events that may effect system resources.
- Risk management includes:
 - risk analysis
 - cost benefit analysis
 - selection of mechanisms
 - implementation and testing
 - security evaluation of safeguards
 - overall security review

DoN Risk Management Process

- Conduct AIS security survey.
 - Develop Activity AIS Security Plan (AAISSL)
 - Issue Interim Authority to Operate
 - Implement minimum mandatory safeguards
- Conduct Risk Analysis
- Develop Security Test and Evaluation Plan
 - Execute tests
- Develop test Contingency Plans
- Compile Accreditation Report
- Issue Accreditation Statement

Risk Analysis

Reasons to Perform Risk Analysis

- Identifies assets and controls.
- Alerts management to near-term risks.
- Pinpoints need for corrective actions.
- Provides guidance for resource expenditures.
- Relates control program to organizational mission.
- Provides criteria for designing and evaluating contingency plans.
- Improves overall awareness.

Steps in doing Risk Analysis

1. Identify and value assets.
2. Determine vulnerabilities.
3. Estimate likelihood of exploitation.
4. Compute expected annual cost.
5. Survey applicable controls.
6. Calculate Return on Investment (ROI).

Risk Analysis

1a. Identify Assets

- Hardware
- Software
- Data
- People
- Documentation
- Supplies

- The following table represents a partial listing of possible assets.

Data Assets	Communications Assets
<i>Classified</i>	
<i>Operations</i>	
<i>Tactical</i>	
<i>Planning</i>	
<i>Financial</i>	
<i>Statistical</i>	
<i>Personal</i>	
<i>Logistic</i>	
<i>Other</i>	
	<i>Communications equipment</i> Communications lines Communications procedures Multiplexors Switching devices Telephones Modems Cables Local area networks

Risk Analysis

Hardware Assets	Software Assets
Central Machine CPU Main Memory I/O Channels Operator's Console	Operating systems Programs Applications Standard applications Test programs Communications Microcomputer
Storage Medium Magnetic Media Disk Packs Magnetic tapes Diskettes Cassettes Drums Non-Magnetic media Punched cards Paper tape Paper printout	Personnel Assets Computer Personnel Supervisory personnel Systems analyst Programmers Applications programmers Systems Programmers Operators Librarian Security officer Maintenance personnel Temporary employees Consultants System evaluator/Auditors Clerical Personnel
Special Interface Equipment Network front ends Database machines Intelligent controllers	Building Personnel Janitors Guards Facility engineers Functional users
I/O Devices User directed I/O devices Printer Card Reader Terminals - local and remote Storage I/O Devices Disk drives Tape drives	Installation Management
Microcomputer Equipment CPU Monitor Keyboard	

Risk Analysis

Administrative Assets	Physical Assets
Documentation	Environmental Systems
Software Hardware File Program JCL System	Air-conditioning Power Water Lighting
Operations	Building
Schedules Operating guidelines Audit documents	Computer Facility Computer room Data reception Tape and disk library Customer engineer room
Procedures	I/O area Data preparation area Physical plant room
Emergency plans Security procedures I/O procedures Integrity controls	Backup equipment Auxiliary power Auxiliary environmental controls Auxiliary supplies
Inventory records	Supplies
Operational Procedures Vital records Priority-run schedule Production procedures	Magnetic media Paper Ribbons
	Office Spaces

Risk Analysis

1b. Valuate Assets

- Hardware
 - What is the replacement cost at current price?
 - How long will it take to replace the system/component?
 - If the work can be done manually, how many more people are required to do the job? How much overtime?
 - If customers contract for services, what are the lost revenues?
- Software
 - How long will it take for a programmer to find the problem?
 - How long will it take to reload and test the program?
 - If it is proprietary software, how long will it take to rewrite the software?
 - If the source code for proprietary software has been disclosed then, what is the probable associated cost?
- Data
 - Can it be replaced?
 - How much will it cost to reconstruct it?
 - Are criminal penalties involved? (police records, tax info, medical info, "Privacy Act" related info)
 - Is the information classified or company confidential? (sales, financial info, product data, weapons research, military operations)
 - Is there a possible loss of life or injury? (life support systems)
- Personnel
 - How many people will have to work overtime?
 - How much will training for the new person cost?
- Difficult to measure
 - Psychological effect (value of customer)
 - Effect of proprietary release (projected sales losses)

Risk Analysis

- When valuating assets it is often convenient to create a value scale.

Scale Number (base 10)	Value in Currency
0	1 or less
1	up to 10
2	11 - 100
3	101 - 1,000
4	1,001 - 10,000
5	10,001 - 100,000
6	100,001 - 1,000,000
7	1,000,001 - 10,000,000
8	10,000,000 +

- Greater resolution can be achieved by using a smaller base.
- The main disadvantage of using scales is that senior management personnel are often trained to think in absolute monetary terms.

Risk Analysis

1b. Estimate Impact Area Value

- Each asset may be viewed as impacting upon one of the three areas of computer security, namely:
 - Secrecy
 - Integrity
 - Availability
- Assigning a dollar value to each of these areas gives a better picture of the asset's overall value
- For example, consider just a simple personnel file which holds personal information about 50 employees. Total loss of the file would require a clerk, earning \$10.00 per hour, three days (24 hours) to reconstruct the file. In addition it might require another clerk to work two hours overtime to process personnel information manually. However, because disclosure of "Privacy Act" information carries a \$10,000 fine, the greatest area of impact is Secrecy

Impact Area	Impact Value
Secrecy	\$10,000
Integrity	N/A
Availability	\$240 + \$90 = \$320

Note:

Navy AIS Security Guidelines uses four impact areas:

- (1) Destruction
- (2) Modification
- (3) Disclosure
- (4) Denial of Service

Risk Analysis

1c. Calculate Total Value for each Asset

$$Value_a = \sum_{i=1}^3 Impact_i$$

$Value_a$ = Impact Area Value (in dollars) for $Asset_a$

$Impact_i$ = Impact Area Value (in dollars) for $Asset_a$

where i represents the three (3) impact areas:

- (1) Secrecy
- (2) Integrity
- (3) Availability

Impact Area	Impact Value
Secrecy	\$10,000
Integrity	N/A
Availability	\$240 + \$90 = \$320
$Value_a = 10,000 + 0 + 320 = 10,320$	

Risk Analysis

2. Determine Vulnerabilities

- Experience, research and imagination all provide help.
 - What are the effects of natural disasters?
 - What are the effects of outsiders?
 - What are the effects of malicious insiders?
 - What are the effects of unintentional errors?

Threats	Threats
<p>Natural</p> <p>Earthquake</p> <p>Flooding</p> <p>Hurricane</p> <p>Landslide</p> <p>Lightning</p> <p>Sandstorm</p> <p>Snow/Ice storm</p> <p>Tornado</p> <p>Tsunami</p> <p>Volcanic eruption</p>	<p>Intentional Acts</p> <p>Bomb threats</p> <p>Disclosure</p> <p>Employee sabotage</p> <p>Enemy overrun</p> <p>Fraud</p> <p>Riot/Civil disorder</p> <p>Strike</p> <p>Theft</p> <p>Unauthorized use</p> <p>Vandalism</p>
<p>Accidents</p> <p>Disclosure</p> <p>Electrical disturbance</p> <p>Electrical interruption</p> <p>Emanation</p> <p>Environmental failure</p> <p>Fire</p> <p>Hardware failure</p> <p>Liquid leakage</p> <p>Operator/User error</p> <p>Software error</p> <p>Telecommunications interruption</p>	

Risk Analysis

3. Estimate Likelihood of Exploitation

- Data from general population.
- Observed data for specific system.
- Estimate number of occurrences in a given time.
- Estimate likelihood from table.
- Delphi Approach
 - several raters compare independent estimates
 - revise until consensus
- Factors affecting threat occurrence:
 - geographic location
 - facility environment
 - proximity to population centers
 - data sensitivity
 - protection/detection features
 - visibility
 - proficiency level
 - security awareness
 - emergency training
 - morale
 - local economic conditions
 - redundancies
 - written procedures
 - compliance level
 - past prosecutions

Risk Analysis

- Frequency of occurrence is normalized based on annual occurrence:

Frequency	Value
Never	0.0
Once in 300 yrs.	1/300 .00333
Once in 200 yrs	1/200 .005
Once in 100 yrs	1/100 .01
Once in 50 yrs	1/50 .02
Once in 25 yrs	1/25 .04
Once in 5 yrs	1/5 .2
Once in 2 yrs	1/2 .5
Yearly	1/1 1.0
Twice a year	2/1 2.0
Once a month	12/1 12.0
Once a week	52/1 52.0
Once a day	365/1 365

Risk Analysis

- The following data is based upon nation statistics and is normalized to annual occurrences.

Threat	Occurrence Rate Range	Threat	Occurrence Rate Range
Natural		Intentional Acts	
Earthquake	.005 - .2	Alteration of data	.083 - .462
Flooding	.01 - .5	Alteration of software	.00225 - .0125
Hurricane	.05 - .5	Bomb threat	.01 - 100
Landslide	0 - .1	Disclosure	.2 - 5
Lightning	.07 - 50	Employee sabotage	.1 - 5
Sandstorm	.01 - .5	Enemy overrun	?
Snow/Ice storm	0 - 10	Terrorist activity	.009 - .10
Tornado	.00001 - 2	Fraud	.09 - .5
Tsunami	0 - .125	Riot/Civil disorder	0 - .29
Volcanic eruption	0 - .01	Theft	.015 - 1
Windstorm	.01 - 10	Unauthorized use	.009 - 5
Accidents		Vandalism	.008 - 1.0
Disclosure	.2 - 5		
Electrical interruption	.1 - 30		
Emanation	.1 - 10		
Environmental failure	.1 - 10		
Fire	.001 - .9		
Hardware failure	10 - 200		
Liquid leakage	.02 - 3		
Operator/User error	10 - 200		
Software error	1 - 200		
Telecommun. failure	.5 - 126		

Risk Analysis

4a. Calculate Annual Loss Expectancy per Threat

$$ALE_t = \sum_{a=0}^n V_a \times O_t$$

ALE_t = Annual Loss Expectancy for $Threat_t$

V_a = Value of $Asset_a$ (0 to n assets)

O_t = Estimated number of occurrences of $Threat_t$ (0 to m threats)

Example

Asset = Server

Threat₁ = Electrical Power Surge

Cost of incident = \$10,000

Event frequency is three (3) times per year

$$ALE_{1,I} = \$10,000 \times 3 = \$30,000$$

Asset = Disk Drive

Threat₁ = Electrical Power Surge

Cost of incident = \$1,000

Event frequency is three (3) times per year

$$ALE_{2,I} = \$1,000 \times 3 = \$3,000$$

Asset = Data Center

Threat₁ = Electrical Power Surge

Cost of incident = \$100,000

Event frequency is three (3) times per year

$$ALE_{3,I} = \$100,000 \times 3 = \$300,000$$

ALE for Threat #1

$$ALE = ALE_{1,I} + ALE_{2,I} + ALE_{3,I}$$

$$ALE = \$30,000 + \$3,000 + \$300,000 = \$333,000$$

Risk Analysis

4b. Calculate Annual Loss Expectancy per Asset

$$ALE_a = \sum_{t=0}^m V_a \times O_t$$

ALE_a = Annual Loss Expectancy for Asset_a

V_a = Value of Asset_a

O_t = Estimated number of occurrences of Threat_t (0 to m threats)

Example

Asset₁ = Data Center

Threat = Electrical Power Surge

Cost of incident = \$100,000

Event frequency is three (3) times per year

$$ALE_{1,1} = \$100,000 \times 3 = \$300,000$$

Asset₁ = Data Center

Threat = Earthquake

Cost of incident = \$1,500,000

Event frequency is once every two years

$$ALE_{1,2} = \$1,500,000 \times .5 = \$750,000$$

Asset₁ = Data Center

Threat = Flood

Cost of incident = \$3,000,000

Event frequency is once every 10 years

$$ALE_{1,3} = \$3,000,000 \times .10 = \$300,000$$

ALE for Asset #1

$$ALE = ALE_{1,1} + ALE_{1,2} + ALE_{1,3}$$

$$ALE = \$300,000 + \$750,000 + \$300,000 = \$1,350,000$$

Risk Analysis

4c. Calculate Total Annual Loss Expectancy

- Determine Total ALE by summing over Threat Categories:

$$ALE = \sum_{t=0}^m ALE_t$$

- Determine Total ALE by summing over all Assets:

$$ALE = \sum_{a=0}^n ALE_a$$

- ALE = Total Annual Loss Expectancy for all asset/threat pairs.

Check for Correctness!

Both calculations of ALE should produce the same value.

Threat/Asset Matrix

	Asset ₁	Asset ₂	...	Asset _n	
Threat ₁	(V ₁ x O ₁) +	(V ₂ x O ₁) +	...	+ (V _n x O ₁)	ALE _{t1}
Threat ₂	(V ₁ x O ₂) +	(V ₂ x O ₂) +	...	+ (V _n x O ₂)	ALE _{t2}
⋮	⋮	⋮	⋮	⋮	⋮
Threat _m	(V ₁ x O _m) +	(V ₂ x O _m) +	...	+ (V _n x O _m)	ALE _{tm}
	ALE _{a1}	ALE _{a2}	...	ALE _{an}	ALE

Risk Analysis

5. Survey New Controls

- Observe which threats produce the greatest ALE_t
- Identify possible controls which may reduce vulnerability (some may apply to several vulnerabilities).
 - Datapro
 - Computer Security Institute Journal
 - Evaluated products listings
 - Other trade journals
 - Consultants

Examples of Controls

Control	Control
surge suppressor	AIS training
Plastic sheets	guard force
PC access control software	backup agreement
cipher locks	access list
smoke/fire detectors	documentation
anti-static mats	peer review
water sensors	off site storage
lightning arrestors	fuse markings
emergency power generator	data labelling
dedicated comm. lines	color coding
halon system	badges/keys
alternative comm. paths	escort procedures

Risk Analysis

6. Project Savings and Compute Return on Investment

- For each control identified:
 1. Identify those vulnerabilities which may be reduced by implementation of the control.
 2. Assign an effectiveness rating for each event/control pair.
 3. Estimate the annual cost of implementing the control.
 4. Calculate the Return on Investment (ROI).

$$ROI = \frac{r_k \times ALE_t}{C_k}$$

C_k = Annual cost for $Control_k$

r_k = Effectiveness rating of $Control_k$

ALE_t = ALE of $Threat_t$

Basis for Selection of Addition Controls
<ul style="list-style-type: none">• Greatest ROI• Minimized ALE

Calculation of ROI

$$ALE = ALE_{1,1} + ALE_{2,1} + ALE_{3,1}$$

$$ALE = \$30,000 + \$3,000 + \$300,000 = \$333,000$$

Threat = Electrical Power Surge

Control = Surge Suppressors (100)

$$C_k = \$50 \times 100 = \$5,500$$

$$r_k = .70$$

$$ROI = \frac{(0.70 \times 333,000)}{(5,500)} = \frac{233,100}{5,500} = 42 : 1$$

Risk Analysis

Quantitative vs. Qualitative Techniques

Quantitative Risk Assessment

- The method we have just described has been the general quantitative approach.
- Most appropriate for large facilities.
- Fundamental problems with the quantitative method:
 - Difficult to find good numbers for threat frequencies.
 - Difficult to estimate value the intangible value of an asset, in particular the “availability” of the information the system was designed to provide.
 - Methodology is essentially incapable of discriminating between low-frequency high-impact threat events (fires) and high-frequency low impact threat events (operator error).
 - Inherent subjectivity of the numbers involved.
 - Labor intensive, time consuming and therefore costly

The Plain Fact Is:

A *truly* quantitative method has not yet been developed!

Qualitative Risk Assessment

- Most appropriate for smaller facilities
- Rather than using pseudo-exact numbers, the qualitative approach uses even fuzzier metrics for asset values, threat frequencies, and control effectiveness:
 - High, Medium, Low
 - One, Two, Three (1,2,3)
 - Vital, Critical, Important, Convenient and Informational
- Advantages
 - Less labor intensive
 - Less Time consuming
- Disadvantages
 - Hard to get support for something with an associated term like “very important” to management
 - Numbers are even more subjective

Risk Analysis

Example Value tables often used in Qualitative Analysis

Financial Loss Table

Financial Loss	Score
Less than \$2,000	1
Between \$2,000 and \$15,000	2
Between \$15,000 and \$40,000	3
Between \$40,000 and \$100,000	4
Between \$100,000 and \$300,000	5
Between \$300,000 and \$1,000,000	6
Between \$1,000,000 and \$3,000,000	7
Between \$3,000,000 and \$10,000,000	8
Between \$10,000,000 and \$30,000,000	9
Over	10

Cost of Disruption Table

Cost of Disruption	Score
Less than \$2,000	1
Between \$2,000 and \$15,000	2
Between \$15,000 and \$40,000	3
Between \$40,000 and \$100,000	4
Between \$100,000 and \$300,000	5
Between \$300,000 and \$1,000,000	6
Between \$1,000,000 and \$3,000,000	7
Between \$3,000,000 and \$10,000,000	8
Between \$10,000,000 and \$30,000,000	9
Over	10

Risk Analysis

Example Value tables often used in Qualitative Analysis

Legal Implications

Extent of Legal Liability	Score
Less than \$10,000	4
Between \$10,000 and \$50,000	5
Between \$50,000 and \$1,000,000 and/or information liable to prosecution.	8
Over \$1,000,000 and/or Senior Officer Liable	10

Breach of Confidentiality

Value to Competitor	Score
Less than \$100,000	4
Between \$100,000 and \$10,00,000	5
Over \$10,00,000	7

Corporate Embarrassment

Extent of Embarrassment	Score
Embarrassment restricted to within the project or work site.	1
Extent of embarrassment spread to other work areas of Operating group or Division	2
Extent of embarrassment spread throughout organization.	3
Public made aware through local press coverage	5
Adverse national press	7

Risk Analysis

Other Techniques

Vulnerability Analysis

- Analyze the vulnerabilities of a department with respect to the people who work in the department.
 - examine each job involved
 - the skills required
 - level of access required
 - working conditions
 - assets which the job impacts

Scenario Analysis

- Useful in visualizing what might happen when no real data is available.
- Highly subjective.
- Valuable in identifying potential threats and vulnerabilities.

Application Risk Assessment

- Identify assets and threats
- Estimate the probabilities and impact of vulnerabilities

	Probability	Exposure	
	Low	Medium	High
Rare	1	3	6
Moderate	2	5	8
High	4	7	9

- Prioritize vulnerabilities
 - Using the chart, locate the priority value on the matrix at the intersection of probability and exposure.
- Identify controls
 - Begin with the highest ranked vulnerabilities, identify potential protective measures.

Risk Analysis

Arguments Against Risk Analysis

- Not Precise
 - values used are imprecise
 - frequency of expected loss are imprecise
 - best used as a planning tool
- False Sense of Precision
 - management often places too much emphasis
 - should emphasize relative sizes
 - an education process
- No Scientific Basis
 - not based on empirical principles
 - based on probability theory/statistical analysis
- Immutability
 - filed and forgotten
 - should be updated annually (Navy says every 3 yrs.)
 - temptation to use previous figures
- Use old figures as a guide:
 - identify changes
 - assets
 - threats/threat frequency
 - replacement costs

Security Plans

Security Plan

A document that describes how an organization will address its security needs.

Three aspects

- What should the plan contain?
- Who should write it?
- How to acquire the support for the plan?

Plan must address the following elements:

- Policy
- Current State
- Recommendations
- Accountability
- Timetable
- Continuing Attention

Recall ==> Risk Analysis Reveals:

- Exposures of greatest potential loss.
- Exposures of greatest expected loss.
- Controls which provide the greatest effectiveness.
- Controls which provide the greatest return per dollar invested.
- The projected savings and feasibility of recommended controls.
- Identify uncovered risks and why they are not covered.
 - control too costly
 - risk insignificant
- What recovery action should be taken.

Security Plans

Responsibility for implementation

- Personal computer users.
- Database administrators.
- Information Officers.
- Personnel staff members.

Timetable

- The order of implementation.
- Measurable milestones for progress assessment.

Continuing Attention

- Establishment of evaluation and review dates.
- Inventory dates for assessing inventory and controls.
- Review of risk analysis.

Members of the Security Planning Team

- Should represent the following groups:
 - Computer hardware group
 - Systems programmers
 - Application programmers
 - Data entry personnel
 - Physical security personnel
 - Representative users
 - Communications personnel

Security Plans

Content of A Security Plan

I. Policy Statement

- Policy statement should specify
 - Organizations security goals.
 - + protect disclosure of information to unauthorized persons
 - + protect integrity of data
 - + protect against loss due to physical disaster
 - Where the responsibility lies.
 - + individuals
 - + groups
 - + managers
 - Commitment (dollars, people, etc.).
- The more precise the policy statement, the easier it will be to interpret and implement.

II. Current Security Status.

- A List of the assets.
- The Security Threat to those assets.
- Controls in place to protect assets.
 - How was this data gathered?
 - How were valuations made?
 - What were the assumptions?

III. Procedure for addressing previously unidentified or new vulnerabilities.

IV. Recommendations.

- Based upon results of risk assessment

Security Plans

Securing Commitment to the Plan

- Acceptability of the plan depends upon:
 - sensibility
 - understandability
 - manageability
- Education can help understandability!
- Management must demonstrate commitment to the plan!

This page is intentionally blank.

This page is intentionally blank.

This page is intentionally blank.

Section 7

Basics of Cryptography

Services Provided by Cryptosystems

Secrecy

- Secrecy requires that an intruder should not be able to determine the plaintext corresponding to given ciphertext, and should not be able to reconstruct the key by examining ciphertext for known plaintext.

Authenticity

- Authenticity requires that the sender can validate the source of a message; i.e., that it was transmitted by a properly identified sender and is not a replay of a previously transmitted message.

Integrity

- Integrity requires the ability to assure that a message was not modified accidentally or deliberately in transit, by replacement, insertion or deletion.

Nonrepudiation

- Protection against a sender of a message later denying transmission.

Introductory Concepts

Definitions

- Encryption - encode.
- Decryption - decode.
- Cryptology - study of encryption and decryption.
- Cryptography - using encryption to conceal text.
- Cryptanalysis - the breaking of secret writing.

- Plaintext - the original message P
 - Sometimes called cleartext.

$$P = [p_1, p_2, \dots, p_n]$$

- Ciphertext - the encrypted message C

$$C = [c_1, c_2, \dots, c_n]$$

Encryption Algorithms

- $C = E(P)$ (E is the encryption algorithm)
- $P = D(C)$ (D is the decryption algorithm)
 - The term encipher is sometimes used for encryption.
 - The term decipher is sometimes used for decryption.

Clearly, we must have:

- $P = D(E(P))$

Substitution Ciphers

The Caesar Cipher

- Each character of the plaintext is replaced with the character three to the right, modulo 26. I.e., A is replaced with D, B is replaced with E, ..., Z is replaced with C.
- $x \bmod y$ (or simply $x \bmod y$) is the remainder obtained when x is divided by y . I.e., $28 \bmod 26 \equiv 2$.
- Modulo is used to handle “wrap around” situations.
- The table below shows how plaintext is encrypted into ciphertext.

p_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Example

p_i	P	R	O	F	E	S	S	I	O	N	A	L		C	O	U	R	T	E	S	Y			
c_i	S	U	R	I	H	V	V	L	R	Q	D	O		F	R	X	U	W	H	V	B			

Variation of Caesar Cipher

p_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

- With any cipher that is a variation of the Caesar Cipher, the message receiver only needs to know what the character A maps to in order to be able to decrypt the whole message. I.e., once you know what character A maps to, you can figure out what all the other characters map to.
- Thus:
 - The *key* of the Caesar Cipher is D.
 - The *key* of the Caesar Cipher variation is P.

Substitution Ciphers

A General Substitution Cipher

- A more general substitution cipher is produced by using a mapping of characters that is not so simplistic as the previous two examples. Consider the mapping below:

p_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	X	D	G	J	M	P	S	V	Y	B	E	H	K	N	Q	T	W	Z	C	F	I	L	O	R	U	A

- In order for the message receiver to decrypt a message using this cipher, they need to know what every character in the alphabet maps to.
- Hence the key needs to be something like:
 - (X,D,G,J,M,P,S,V,Y,B,E,H,K,N,Q,T,W,Z,C,F,I,L,O,R,U,A).

Cryptanalysis Attack of These Codes:

- Since there are only 26 different keys for Caesar Cipher variation codes, one could try all keys in an attempt to decrypt a message.
- This type of an attack (trying all keys) is called a brute-force attack.
- There are $26!$ (26 factorial = $26 \times 25 \times 24 \times \dots \times 1$) keys for the general substitution cipher example. $26! \cong 4 \times 10^{26}$
- This number of keys is too great to attempt a brute force attack.
- In spite of this, this type of cipher is easy to crack.
- Letter frequency analysis is commonly used to break substitution ciphers.

Frequency Distribution Analyses

Frequency Distribution of Characters in English:

- The following tables list the relative frequency of characters in the English language.

Character	E	T	R	N	I	O	A	S	D	L	H	C	F
percent	12.75	9.25	8.50	7.75	7.75	7.50	7.25	6.00	4.25	3.75	3.50	3.50	3.00

Character	U	M	P	Y	G	W	V	B	K	X	Q	J	Z
percent	3.00	2.75	2.75	2.25	2.00	1.50	1.50	1.25	0.50	0.50	0.50	0.25	0.25

Consider the following ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFAPPDTSPVQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- The distribution of characters in this message is:

Character	P	Z	S	U	O	M	H	D	E	V	X	F	W
percent	13.13	11.67	8.33	8.33	7.50	6.67	5.83	5.00	5.00	4.17	4.17	3.33	3.33

Character	Q	T	A	B	G	Y	I	J	C	K	L	N	R
percent	2.50	2.50	1.67	1.67	1.67	1.67	0.83	0.83	0.00	0.00	0.00	0.00	0.00

- It seems likely that cipher letters P and Z are the equivalents of plaintext letters E and T, but it is not certain which is which.
- The letters S, U, O, M and H are all of high frequency and probably correspond to plaintext letters from the set {R, N, I, O, A, S}.

Frequency Distribution Analyses

Additional Strategies:

- The frequency of two-letter combinations (known as digraphs) can also provide clues.
- For example, the digraph ZW appears three times.
- The most common digraph is “TH”.
- Also ZWP appears in the ciphertext and we conjectured that P might stand for E in plaintext.
- Furthermore, ZWSZ appears in the first line.
- It is possible that S stands for A.
- Given these assumptions we have the following structure:

U	Z	Q	S	O	V	U	O	H	X	M	O	P	V	G	P	O	Z	P	E	V	S	G	Z	W	S
t		a										e		e		t	e		a		t	h	a		

Z	O	P	F	P	E	S	X	U	D	B	M	E	T	S	X	A	I	Z	V	U	E	P	H	Z	H
t		e		e	a									a			t				e		t		

M	D	Z	S	H	Z	O	W	S	F	P	A	P	P	D	T	S	V	P	Q	U	Z	W	Y	M	X
		t	a		t	h	a		e		e	e			a	e			t	h					

U	Z	U	H	S	X	E	P	Y	E	P	O	P	D	Z	S	Z	U	F	P	O	M	B	Z	W	P
t		a		e		e		e		e		t	a	t			e			t	h	e			

F	U	P	Z	H	M	D	J	U	D	T	M	O	H	M	Q
		e	t												

- At this point, trial and error should yield the plaintext.

More On Substitution Ciphers

Monoalphabetic Substitution Ciphers:

- All ciphers discussed so far are examples of monoalphabetic ciphers.
- Throughout the whole message, each character of plaintext is always replaced by the same character of ciphertext.
 - For example, when using the Caesar Cipher, the plaintext letter I is always replaced with the ciphertext letter L.
- Any cipher that has this property is called a monoalphabetic cipher.
- If the message is long enough, the distribution of letters in the ciphertext will be similar to the distribution of letters in English.
 - If the letter E occurs 13% of the time in the plaintext, then the letter that E encrypts to will occur 13% of the time in the ciphertext.
- Thus, monoalphabetic ciphers lend themselves to character frequency analyses and are relatively easy to break.

Advantages of substitution ciphers:

- Can be performed by direct lookup.
- Time to encrypt a message of n characters is proportional to n .

Polyalphabetic Ciphers

Polyalphabetic Cipher Issues:

- Polyalphabetic Ciphers are an improvement over the simple monoalphabetic technique.
- Polyalphabetic Ciphers use two or more monoalphabetic ciphers when encrypting a message.

A Simple Polyalphabetic Cipher

- Consider an example which uses a variation of the Caesar Cipher with key D on even letters and key M on odd letters.
- In this example the letter E in the plaintext sometimes encrypts to the letter H (when E is in an even position in the plaintext) and sometimes encrypts to letter Q (when E is in an odd position in the plaintext).
- The resulting ciphertext will not then exhibit the same frequency distribution of characters as the plaintext.
- If E occurs 13% of the time in the plaintext, there may be no character in the ciphertext that occurs 13% of the time, because sometimes E is mapped to H and sometimes it is mapped to Q.
- This example cipher is still relatively easy to break.
- After failing to break a message using a straight forward frequency analysis, the cryptanalyst might assume that the cipher is a polyalphabetic cipher and might start looking at frequency distributions of every other letter or every third letter or every fourth letter and so on.
- In the case of this example, if the message is long enough or if enough messages have been intercepted, a frequency analysis of every other letter would break the code.

Advantages of polyalphabetic ciphers:

- Flattens letter frequencies.
- Double letter pairs not so obvious.

Vigenère Cipher

Vigenère Cipher Details:

- The Vigenère Cipher is one of the best known polyalphabetic ciphers.
- Consider the following example:

key	m	o	n	t	e	r	e	y	m	o	n	t	e	r	e	y	m	o	n	
plaintext	w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	
ciphertext	I	S	N	K	I	U	M	Q	O	C	I	X	V	V	H	Q	M	J	R	R

- In this example the key used is the word “monterey”.
 - Note how the word “monterey” is written repeatedly for the whole length of the message in the top row of the table.
- The key specifies which variation of the Caesar Cipher is used for each letter of the message.
 - For example, the first letter of the message will be encrypted with a Caesar Cipher variation of key M.
 - The second letter will be encrypted with a Caesar Cipher variation of key O.
 - An so on.

Attacking this type of cipher.

- After failing to break a message using a straight forward frequency analysis, the cryptanalyst might assume that the cipher is a polyalphabetic cipher and might start looking at frequency distributions of every other letter or every third letter or every fourth letter and so on.
- In this example, if the message is long enough or if enough messages have been intercepted, a frequency analysis of every eighth letter would break the code.

One Time Pad Scheme

How To Make An Unbreakable Cipher

- If in the previous example, the key word was random and as long as the message, how would a cryptanalyst attack this cipher?
- The cryptanalyst would need to intercept many messages to develop a statistical relationship between the ciphertext and the plaintext.
 - I.e., the first letter of each message would always be encrypted using the same variation of the Caesar cipher, as would the second letters and so on.
- If enough messages are intercepted, the code could be broken.
- If each random key is only used once (no two messages use the same random key), how would a cryptanalyst attack this cipher?
- No successful attack is possible because the ciphertext has no statistical relationship to the plaintext.
- This type of cipher is called a One Time Pad and it is unbreakable.
- Note however, that this scheme requires the secure distribution of many long (as long as the messages) keys.

Binary Substitution Ciphers

The Vernam cipher:

- The Vernam cipher is a version of the One Time Pad cipher that is implemented using binary keys, plaintext and ciphertext.
- Consider the example below:

key	1	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	1	0	1	0	0	0
plaintext	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	1	1	0	1
ciphertext	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	0	1	0	0	1

- The ciphertext is obtain by XORing key bits with plaintext bits.
- I.e., $C_i = P_i \oplus K_i$
- If the key is not as long as the entire message and is therefore repeated, a statistical relationship will exist between the plaintext and the ciphertext and the cipher may be broken.
- If the key is as long as the message, but the key is used for several messages, there will again be a statistical relationship between the plaintext and the ciphertext so that the cipher may be broken.
- If the key is as long as the message and only one message is encrypted with any one key, the code is functionally equivalent to a One Time Pad and is unbreakable.

Transposition Techniques

Transposition Ciphers:

- All examples so far involved the substitution of a ciphertext symbol for a plaintext symbol.
- A very different kind of mapping is achieved by performing a permutation of the plaintext letters.
- Pure transposition ciphers are easily recognized because they have the same letter frequencies as the original plaintext.

Example 1:

Writing the message backwards - not very hard to analyze.

- The plaintext “thetimehascomethewalrussaid”
- The ciphertext “diassurlawehatemocsahemiteht”

Example 2:

Transposing adjacent letters - not very hard to analyze.

- The plaintext “thetimehascomethewalrussaid”
- The ciphertext “httemihesaocemhtwelausrssiad”

Example 3:

Write the message down in columns and reading off the rows becomes the ciphertext.

- The plaintext “thetimehascomethewalrussaid”

t	e	i	e	a	c	m	t	e	a	r	s	a	d
h	t	m	h	s	o	e	h	w	l	u	s	i	

- The ciphertext “teieacmtearsadhtmhsoehwlusi”
- Cryptanalysis is fairly straightforward and involves laying out the ciphertext in matrices of various shapes and sizes.

More Transposition Techniques

Example 4:

- Again the plaintext is written down by column and it is read off by rows, but this time the rows are read off in a permuted order.
 - The key column specifies the order in which the rows are read off.
- The plaintext “thetimehascomethewalrussaid”

key						
3	t	m	c	h	r	i
1	h	e	o	e	u	d
4	e	h	m	w	s	
5	t	a	e	a	s	
2	i	s	t	l	a	

- The ciphertext “heoeudistlatmchriehmwstaeas”
- The cipher is still fairly easy to break, by playing around with different permutations of rows and columns.
- Digraph (common two-letter combinations) and trigraph (common three-letter combinations) frequency tables can be useful.

Multiple Stage Ciphers:

- Transposition ciphers can be made significantly more secure by performing more than one stage of transposition.
- The result is a more complex permutation that is not easily reconstructed.
- Ciphers consisting of multiple stages of transpositions and multiple stages of substitutions can be very secure.

Encryption Issues

Data Compression:

- Compressing the message before encrypting it can enhance a cipher's ability to resist being broken.
- Many of the cryptanalytic techniques discussed so far involved making guesses and then looking for English words.
- If the message is compressed before encryption, it does not look like English when it is correctly decrypted (it must be uncompressed to recover the English text).

Types of attacks on ciphers:

- Often today it is assumed that the adversary knows what encryption algorithm is being used.
- When this is true, the adversary is only attempting to determine the key used during the encryption process.
- Cryptographers try to determine the strength of ciphers given that a cryptanalyst may possess different types of information.
- Possible scenarios include:
 - Ciphertext only
 - The cryptanalyst knows the algorithm and the ciphertext.
 - Known plaintext
 - The cryptanalyst knows the encryption algorithm and a plaintext-ciphertext pair (the plaintext that corresponds to a ciphertext).
 - Somehow the cryptanalyst has obtained the plaintext corresponding to a ciphertext.
 - Chosen plaintext
 - The cryptanalyst knows the encryption algorithm and a plaintext-ciphertext pair, such that the plaintext was chosen by the cryptanalyst.
 - Somehow the cryptanalyst has tricked someone into sending a message that might reveal information or structure about the key being used.

More Encryption Issues

Computational Security:

- An encryption scheme is said to be computationally secure if
 - the cost of breaking the cipher exceeds the value of the encrypted information and
 - the time required to break the cipher exceeds the useful lifetime of the information.

Abstract measures of a cipher's effectiveness:

- Confusion:
 - Confusion obscures the relationship between the plaintext and the ciphertext.
 - The easiest way to do this is through substitution.
- Diffusion:
 - Diffusion dissipates the redundancy of the plaintext by spreading it out over the ciphertext.
 - The simplest way to cause diffusion is through transposition.
- *Confusion and diffusion are the cornerstone of good block cipher design.*
- Bit-sensitivity
 - Bit-sensitivity looks at the impact on the ciphertext (how many bits change) as a result of changing either one bit of the plaintext or of the key. We want every bit of the ciphertext to depend on every bit of the plaintext and on every bit of the key.

Codes based on hard problems:

- Just because a cipher is based on a “hard problem” it does not mean that the cryptanalyst needs to solve that problem to break the code.
- Recall that the General Substitution Cipher has $26!$ keys, which is far too many to try exhaustively.
- But this cipher is easily broken using a frequency distribution analysis of the ciphertext.
- Brute force attacks are usually impractical.

The Data Encryption Standard (DES)

History of DES

1972 - NBS issued a call for proposals:

- Must provide high level of security.
- Must be completely specified and easy to understand.
- The security of the algorithm must reside in the key; the security should not depend on the secrecy of the algorithm.
- Must be available to all users.
- Must be adaptable for use in diverse applications.
- Must be economical to implement in electronic devices.
- Must be efficient.
- Must be able to be validated.
- Must be exportable.

1974 - IBM responded with “Lucifer” (renamed - DEA).

- Note that Lucifer algorithm used a 128-bit key and DES uses a 56-bit key.
- IBM consulted NSA on design issues.
 - NSA suggested changes to some of the S-boxes.

1976 - DES officially adopted.

Overview of DES

- Combination of:
 - Substitution technique (for confusion).
 - Transposition technique (for diffusion).
- These two techniques are repeated for 16 cycles one on top of the other.
- Plaintext is encrypted in blocks of 64 bits.
- Keys are 64 bits long (only 56 are really needed).
- Uses only standard arithmetic and logical operations on up to 64 bit numbers.

More on DES

DES Has Four Modes of Operation

- ECB -Electronic Code Book
- CBC - Cipher Block Chaining
- OFB - Output Feedback
- CFB - Cipher Feedback
- More will be said about the differences between these modes of operation later.

The following description of the DES algorithm will assume the ECB mode of operation.

When used for encryption:

- Data is input in a block, which consists of 64 bits.
- A 64 bit key is input.
 - Only 56 bits of the key are used.
 - Every 8th bit is discarded.
 - The extra bits can be used as parity-check bits to ensure the key is error free.
- A 64 bit block of ciphertext is output.

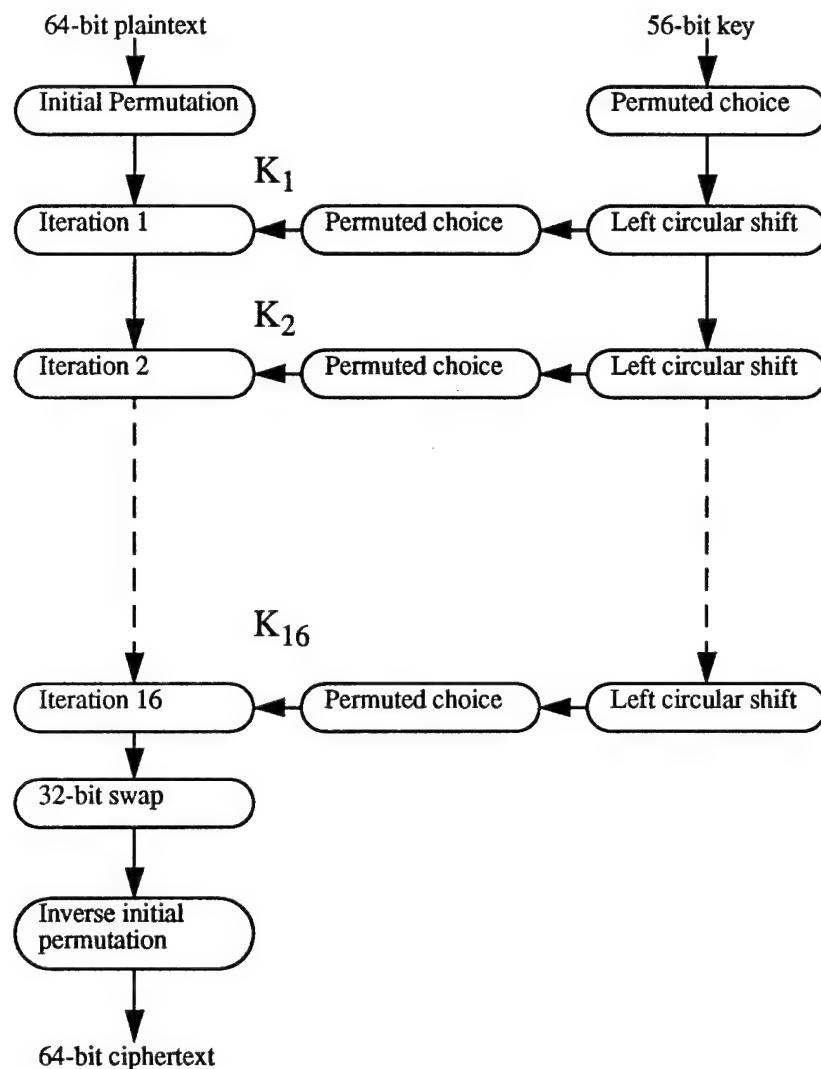
When used for decryption:

- A 64 bit block of ciphertext is input
- The same key used during encryption is input.
- A 64 bit block of plaintext is output.

Basic algorithm structure

- The figure on the following page reveals the basic algorithm structure.
- The algorithm has 16 iterations.
- The Key goes through 16 transformations.

Overall DES Scheme

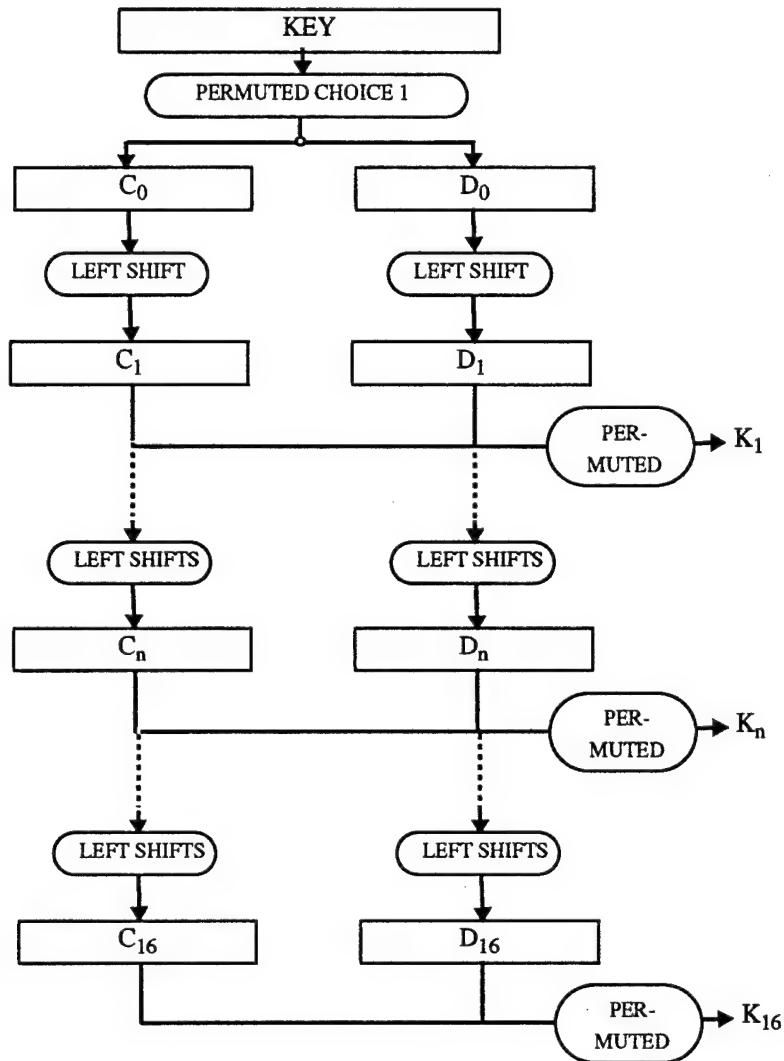


Internal DES Details

DES Structure

- Notice the two separate lines of processing in the figure on the previous page.
 - An encrypting algorithm on the left.
 - A key transforming algorithm on the right.
- The following two pages discuss the inner structure of the key transforming algorithm.
 - This processing produces the various key values (K_1 through K_{16}) that are used by the encrypting algorithm.
- Following the discussion of the key transforming algorithm is a discussion of the plaintext encrypting portion of the algorithm.

DES Key Transforming Algorithm



DES Key Transforming Algorithm

The Key Scheduler

- Contains a set of bit-shifts and permutations totally independent of the encrypting algorithm.
- The key schedule is usually computed before encrypting takes place.

Step 1

1. The key is subjected to an initial Permuted Choice P-Box.
2. The result is divided into two 28-bit halves labeled C_0 and D_0 .

Step 2

1. Both C and D are given a left circular shift according to the shift table.

Step 3.

1. C and D are concatenated to produce CD_1 .

Step 4

1. CD_1 is then subjected to a Permuted Choice in which the key is permuted.
2. Bits 9, 18, 22, 25, 35, 38 and 54 are removed to produce a 48-bit key K_1 .
3. K_1 is used in cycle 1 of the encryption algorithm.

Steps 2 through 4

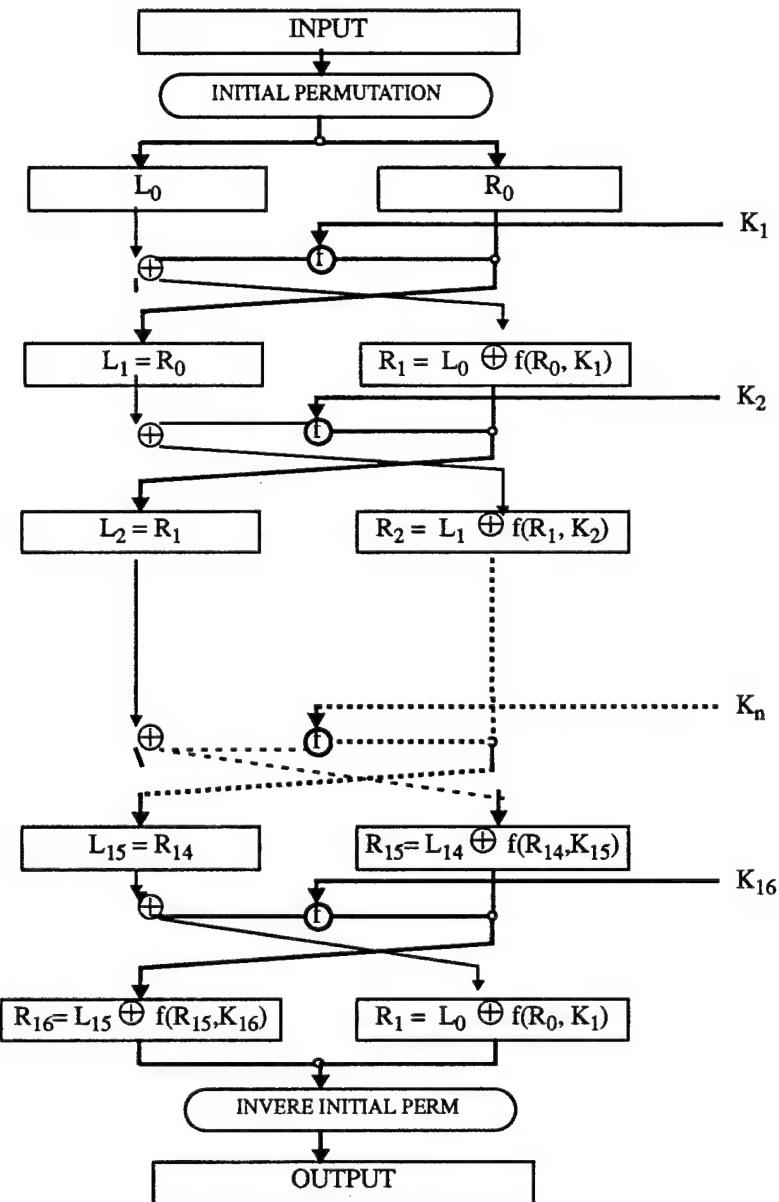
1. Repeated a total of 16 times.

Note:

The only difference in each cycle is the number of bits shifted in the circular shift.

DES Encrypting Algorithm

Below is a block diagram of the plaintext encryption algorithm.



DES Encrypting Algorithm

A Single Cycle of the DES

Step 1.

1. Input 64 bits of plaintext.

Step 2.

1. Rearrange by a P-Box known as the Initial Permutation (IP).

Step 3.

1. Split block into two 32 bit segments called the Left (L) and Right (R) halves.
2. Save a copy of the Right half and label R_0 .

Step 4.

1. Subject R to a special permutation box called a Permutation Expansion (P_E) which takes 32 input bits and produces 48 output bits.

Step 5.

1. Take the expanded R and XOR it against a 48-bit segment of the key.
(Note: This is the only place in each cycle which involves the key)

Step 6.

1. Output from XOR is called Pre-S block.
2. 48-bit Pre-S block is broken into eight 6-bit segments.
3. Each segment processed by a different S-Box in parallel.
4. Each S-Box produces four bits.
5. A total of 32 bits output called Post-S.

Step 7.

1. Post-S is fed to a final P-box.
2. Takes a 32-bit input and returns a 32-bit output.
3. The output is called the result.

DES Encrypting Algorithm

Note:

Steps 4 through 7 are often grouped into one function in DES diagrams $f(R_{n-1}, K_n)$

Step 8.

1. The Left (L) is now XORed against the output of $F(R, K_n)$ to produce the “New R”
2. R_0 now becomes the “New L”

Steps 3 to 7

1. repeated 16 times for each 64-bit block to be encrypted

- Finally, the PREOUTPUT is subjected to a reverse of the initial permutation (IP)
- This is required for the algorithm's invertibility.
- Decryption uses the exact same algorithm except that the order in which the keys are used is reversed.

Substitution Boxes (S-Boxes)

Substitution Box (S-Box)

- Introduces confusion and non-linearity to DES
- Interpret bits as numbers
- One number replaced by another from a table
 - table has values ranging from 0 (0000) to 15 (1111)
 - duplications among the elements
- Takes 6-bit input and returns 4-bit output
 1. First and last bits choose row into S-box substitution table.
 2. The middle four bits chooses the column
 3. The table returns a four bit number
- They are the heart and soul of the algorithm's secrecy

S-Box

	Column Number															
Row No	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Example S-box Input/Output

INPUT	binary 101011 = decimal 43
First and Last bits	binary 11 = decimal 3
Middle four bits	binary 0101 = decimal 5
OUTPUT	binary 1001 = decimal 9

ECB Mode

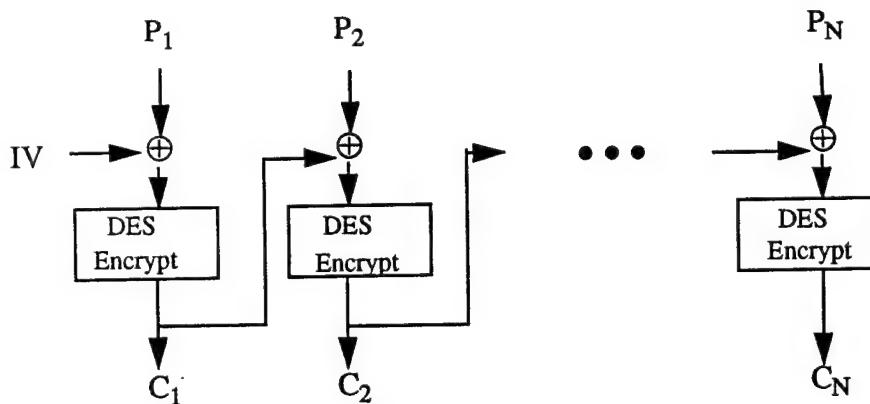
Electronic Codebook Mode (ECB)

- Each 64-bit plaintext block is encrypted independently of all other plaintext blocks.
- The term codebook is used because, for a given key, there is a unique ciphertext for every 64-bit block of plaintext.
- Abstractly, one could imagine a gigantic codebook with an entry for every 64-bit plaintext block and the corresponding 64-bit ciphertext block.
- If a message is highly structured, it may contain blocks of plaintext that are identical. And since this mode encrypts them to identical ciphertext blocks, some structure of the message maybe revealed.
- Hence, this mode is not considered too secure for long messages.
- An advantage of this mode is that due to the independence of the block encryptions, an error that occurs during transmission in one block will only affect the decryption of that block.
 - I.e., errors do not propagate.

CBC Mode

Cipher Block Chaining Mode (CBC)

- An enhanced version of the ECB that chains together blocks of ciphertext.
- The CBC mode encrypts each block using the plaintext, the key and the output of the previous block (except the first in the cycle which uses an Initializing Vector (IV)).
- The CBC mode has an advantage over the ECB mode in that repeating blocks are hidden.
 - See the diagram below.
- The CBC is frequently employed in generating Message Authentication Codes (MACs), frequently referred to as Message Digests, which are a type of cryptographic checksum used to ensure message integrity.
 - The MAC consists of the last block of ciphertext and is generally sent along with a plaintext version of the message.
 - The rest of the ciphertext message is discarded when all that is desired is a checksum (MAC).

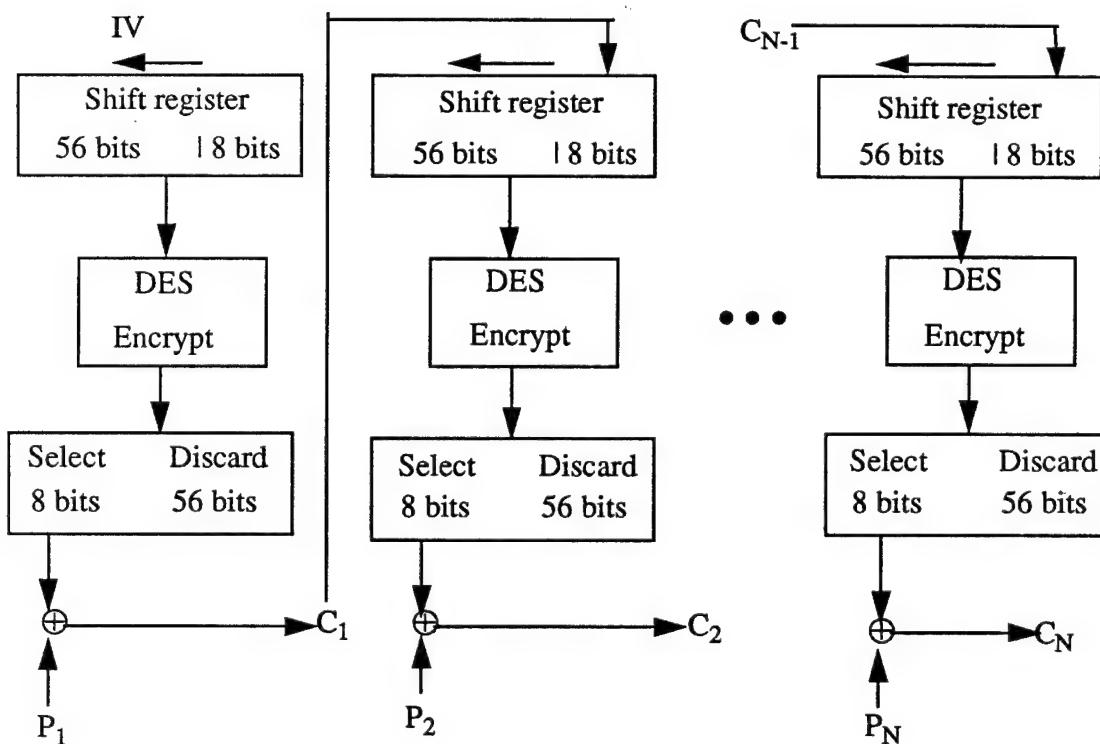


- Because of the chaining effect on blocks, an error in transmission will cause decryption errors in subsequent blocks.

CFB Mode

Cipher Feedback Mode (CFB)

- This mode uses the block nature of DES in a way that produces a stream cipher.
- Stream ciphers act on small chunks of data, usually 8-bit chunks.
- It eliminates the need to pad messages into 64-bit blocks.
- It can operate in real-time. That is, each character can be encrypted and transmitted immediately.
- The scheme requires an Initial Vector (IV) to start the process.
- In the diagram below, P_i is an 8-bit piece of plaintext and C_i is the corresponding 8-bit piece of ciphertext.
- Notice that the plaintext never gets directly processed by the DES Encryption algorithm. Instead, it is XORed with the output of the DES Encryption algorithm.



OFB Mode and Decryption

Output Feedback Mode (OFB)

- This mode is similar in operation to the CFB mode.
- The difference is that in CFB mode the previous 8-bit chunk of ciphertext is shifted into the shift register and used as input to DES and in OFB mode the selected 8-bits of DES output are shifted into the shift register.
- One advantage of OFB mode is that errors in transmission do not propagate.

DES Decryption

The process of decryption with DES is essentially the same as the encryption process.

The same algorithm and key are used for encryption and decryption, except that during decryption the internal keys (K_i) are used in reverse order.

DES is a form of shared key / symmetric key cryptography.

DES Issues

Criticisms of the DES

- Number of iterations - is 16 enough?
- Key length
 - 2^{56} possible keys to try.
 - Massively parallel system could try all keys in 1 day (although it would be a very expensive proposition).
 - *According to an article in the October 1, 1996 San Jose Mercury News, a government agency can break a DES encrypted message in 12 seconds.*
 - Triple encryption may be the answer. See below.
- NSA involvement - Do they hold a 'trapdoor'?

Weaknesses of the DES

- Weak keys (e.g. all zeros or all ones).
- Semi-Weak keys (2 separate keys can decrypt the same message).
- The same DES algorithm is used!
- Key length

Triple DES

- Provides an effective key length of 112 bit key (i.e. independent 56 bit keys); thereby making a brute force attack infeasible.
- Most common variant is EDE mode (encrypt-decrypt-encrypt for encryption and decrypt-encrypt-decrypt for decryption).
 - Encrypt plaintext with DES using key #1.
 - Decrypt resulting cipher text with key #2.
 - Encrypt resulting Ciphertext with key #1.
- About half as fast as standard DES
- The keys are used in reverse order.

Bit Sensitivity

- If only one bit of either the input plaintext or key is changed, each bit of the ciphertext is affected.

Symmetric Versus Asymmetric Algorithms

Two Categories of Encrypting Algorithms

- Encryption algorithms can be divided into two categories:
 - Symmetric key algorithms or ciphers
 - Asymmetric key algorithms or ciphers

Symmetric Key Ciphers:

- In symmetric key ciphers, both the encryption algorithm and the decryption algorithm use the same key.
- All ciphers discussed so far, including DES, are symmetric key algorithms.
- Other names for symmetric ciphers:
 - Private key.
 - Secret key.
 - Single key.
 - Shared key.
 - Conventional encryption.
- Symmetric key schemes require both the sender and receiver to possess the same key.
 - I.e., the key must be securely distributed.
- The amount of information a cryptanalyst can gain about a key is directly proportional to the number and length of messages encrypted with the key.
- For security reasons, keys should be changed periodically, which means that keys need to be securely distributed fairly often.

Asymmetric Key Ciphers:

- In asymmetric key ciphers the key used for encryption is different from the key that is used for decryption.
- Other names for asymmetric ciphers:
 - Public key.
- Asymmetric key ciphers do not require the secure distribution of keys.
 - They, however, have other key distribution problems.
 - These problems are discussed on future slides.

International Data Encryption Algorithm (IDEA)

Overview

- Operates on 64-bit plaintext block.
- Uses 128 bit key.
- Same algorithm is used for encryption and decryption (like DES).
- Considered by some to be superior to DES
- It is a symmetric algorithm.

General Description

- 64 bit input block is divided into four 16 bit blocks: X1, X2, X3, and X4 which become the input blocks to the first round of the algorithm.
- In each of the eight total rounds, the four sub-blocks are XORed, added, and multiplied with one another and with six 16 bit sub-blocks of key material.
- Between each round the second and third sub-blocks are swapped.

Speed of IDEA

- Software implementation speeds are comparable with those for DES.
- Hardware implementations are just slightly faster.

Want to know more?

If you are interested in learning more about IDEA and other cryptographic techniques then you might want to read the following book:

*Applied Cryptography
Protocols, Algorithms, and Source Code in C,
Second Edition
by
Bruce Schneier*

Skipjack Algorithm

Overview

- Developed by NSA.
 - started design in 1985 and finished evaluation in 1990
- Developed for use by Clipper and Capstone.
- Actual algorithm is classified SECRET.
 - To prevent the construction of devices that will interoperate with Skipjack devices, but which don't support the "Law Enforcement Field" mechanisms.

General Description

- It is a symmetric algorithm.
- It has an 80 bit key and encrypts 64-bit blocks of plaintext.
- It can be used in either ECB, CFB, OFB, or CBC modes.
- There are 32 rounds of processing per single encrypt or decrypt operation.
- The strength of Skipjack does not merely depend upon the secrecy of the algorithm (like any good cipher).

Speed of Skipjack

- The algorithm was designed to achieve high data throughput for use in real-time communications system.

Skipjack Issues:

- It is intended to only be implemented in a tamper-proof chip.
- It is also intended that the implementation will provide a Law Enforcement Field (LEAF) that will enable law enforcement agencies to decrypt encrypted messages.

Public-Key Cryptography

Asymmetric Key Cryptography

- Public key cryptographic systems use two keys, one private and one public key, to make the necessary transformations.

Summary of Public Key Protocols

- Each user generates two keys - a public key and a private key.
- Each user keeps the private key in a secure manner.
- Each user gives the public key to everyone else.

Example for sending a secret.

- Alice wants to send a message to Bob, such that if the message is intercepted it cannot be read.
- Alice has Bob's public key. (Only Bob has Bob's private key!)
- Alice encrypts the message with Bob's public key.
- Alice sends the encrypted message to Bob.
- Bob uses his private key to decrypt the message.
- If the message is intercepted in transit, it can only be decrypted by someone who has Bob's private key and no one but Bob has Bob's private key.

RSA Encryption

Rivest-Shamir-Adelman (RSA) Encryption

- Two key system (public and private) based on the difficulty of factoring very large numbers.
 - Encryption $C \equiv P^e \pmod{n}$
 - Decryption $P \equiv C^d \pmod{n}$
- Key_e and Key_d are carefully chosen such that:

$$P \equiv (P^e)^d \pmod{n} \equiv (P^d)^e \pmod{n} \text{ (i.e. } E(D(M)) = D(E(M)) = M\text{)}$$

Choosing Keys for RSA Method

- Underlying problem is based on factoring very large numbers!
Encryption $C \equiv P^e \pmod{n}$
Decryption $P \equiv C^d \pmod{n}$
where
Encryption key = (e, n)
Decryption key = (d, n)
- The first task is to select n
 - n is normally very large (approx 200 digits)
 - n is a product of two large primes p and q (typically 100 digits each)
- Next a large integer e is chosen such that
 - e is relatively prime to $(p-1) * (q-1)$
 - I.e., e and $(p-1) * (q-1)$ have no factors in common.
 - e is usually picked as a prime larger than both $(p-1)$ and $(q-1)$
- Next select d such that: $e * d \equiv 1 \pmod{(p-1) * (q-1)}$
- Then if we have selected our numbers correctly,

A MINOR MIRACLE OCCURS

$$(P^e)^d \equiv P \pmod{n}$$

Note: Minor miracle provided courtesy of Euler and Fermat.

RSA Encryption

Summary of RSA Encryption

Public Key

$n \Rightarrow$ product of two primes, p and q
(p and q remain secret)
 $e \Rightarrow$ relatively prime to $(p-1) * (q-1)$

Private key

$$d \equiv e^{-1} \pmod{(p-1) * (q-1)}$$

Encrypting

$$c \equiv m^e \pmod{n}$$

Decrypting

$$m \equiv c^d \pmod{n}$$

Example: RSA Encryption

Let $p = 11$ and $q = 13$ {both primes}

Then $n = p * q = 143$ and
 $(p-1) * (q-1) = 10 * 12 = 120$

Next choose e such that it is relatively prime to
 $(p-1) * (q-1)$. We will choose 11!

Recall that: $d \equiv e^{-1} \pmod{(p-1) * (q-1)}$
 $d * e \equiv 1 \pmod{(p-1) * (q-1)}$

In other words: $11 * 11^{-1} \pmod{120} \equiv 1 \pmod{120}$
 $121 \pmod{120} \equiv 1 \pmod{120}$

In this case both e and d are the same (11)

Let the plain message P be the letter 'H'
(7 in our 0-25 schema)

$$\begin{aligned} E('H') = E(7) &= 7^{11} \pmod{143} \equiv 106 \\ D(106) &= 106^{11} \pmod{143} \equiv 7 \Rightarrow 'H' \end{aligned}$$

Introduction to Hash Functions

Hash Functions and Message Digests.

- A hash function H accepts a variable-size message M as input and outputs a fixed-size representation $H(M)$ of M , sometimes called a message digest. In general $H(M)$ will be much smaller than M ; e.g., $H(M)$ might be 64 or 128 bits, whereas M might be a megabyte or more.
- A hash function can serve to detect modification of a message. That is, it can serve as a cryptographic checksum (also known as an MDC = manipulation detection code or MAC = message authentication code).
- It is theoretically possible that two distinct messages could be compressed into the same message digest (a collision). The security of hash functions thus requires collision avoidance. Collisions cannot be avoided entirely, since in general the number of possible messages will exceed the number of possible outputs of the hash function. However, the probability of collisions must be low.

Properties of Hash Functions

- To serve the authentication process properly a hash function F must have the following properties:
 - a. F can be applied to an argument of any size.
 - b. F produces a fixed-size output.
 - c. $F(x)$ is relatively easy to compute for any given x .
 - d. For any given y it is computationally infeasible to find x with $F(x) = y$.

Property (d) guarantees that an alternative message hashing to the same value as a given message cannot be found. This prevents forgery and also permits F to function as a cryptographic checksum for integrity.

Example Hash Function

Ultra simple example:

- Divide the message up into 8-bit chunks.
- Pad the beginning with zeros if necessary.
- Interpret each 8-bit block as a number.
- The values will be between 0 and 255.
- The message M is $M_1, M_2, M_3, \dots, M_N$
- The hash of message M ($\text{hash}(M)$) $\equiv (M_1^2 + M_2^2 + M_3^2 + \dots + M_N^2) \bmod 255$
- $M = 11001010, 00101010, 01011100$
- $\text{hash}(M) \equiv ((202)^2 + (42)^2 + (92)^2) \bmod 255$
- $\text{hash}(M) \equiv (40804 + 1764 + 8464) \bmod 255$
- $\text{hash}(M) \equiv (51032) \bmod 255$
- $\text{hash}(M) = 32$
- How hard is it to find another message that also hashes to 32?

Digital Signature Algorithm (DSA)

The Digital Signature Algorithm.

- The scheme relies on the difficulty of computing discrete logarithms.
- It was proposed in 1991 as the basis for the NIST Digital Signature Standard (DSS).
- It is intended to provide the capability for the creation and verification of digital signatures and not for use as a general encryption algorithm nor for key distribution.
- DSA is notably slower than RSA (10 to 40 times slower for signature verification).
- It is the subject of controversy since it may infringe upon other patents.
- It has been criticized because of its short key length.
- In 1992 Simmons discovered a subliminal channel in DSS (which allows people to embed a secret message in their signatures that can only be read by the receiver of the message).

Summary of DSA Signatures

Public Key

$p \Rightarrow$ 512-bit to 1024-bit prime (can be shared among a group of users)

$q \Rightarrow$ 160-bit prime factor of $p-1$ (can be shared among a group of users)

$g \Rightarrow < p$ (can be shared among a group of users)

$y \Rightarrow g^x \pmod{p}$ (a 60-bit number)

Private key

$x = < q$ (a 160-bit number)

Signing

$k \Rightarrow$ choose at random, less than q

r (signature) $\equiv g^k \pmod{p} \pmod{q}$

s (signature) $\equiv (k^{-1} (H(m) + xr)) \pmod{q}$

Verifying

$w \equiv s^{-1} \pmod{q}$

$u1 \equiv (H(m) * w) \pmod{q}$

$u2 \equiv (r * w) \pmod{q}$

$v \equiv ((g^{u1} * y^{u2})) \pmod{p} \pmod{q}$

Accept as valid if $v = r$

MD4/5 and MD2

MD4/MD5 and MD2

- MD4 is a one-way hash function designed by Ron Rivest. MD stands for Message Digest, and the algorithm produces a 128-bit hash, or message digest, of the input message.
- Rivest's goals for the design of MD4 algorithm were:
 - Security - It should be computationally infeasible to find two messages that hash to the same value. No attack should be more efficient than brute force.
 - Direct Security - Not to be based on any fundamental assumption like the difficulty of factoring.
 - Speed - Should be suitable for high speed 32 bit software implementations.
 - Simplicity - Should be as simple as possible without large data structures.
 - Architecture - favor microprocessor architectures (specifically Intel microprocessors).
- After a successful attack was made on the first three rounds of the algorithm was achieved in 1990, Rivest strengthened the algorithm which is now known as MD5 (MD5, to date is considered secure).
- MD2 is another one-way hash function also created by Rivest and is used as the basis for Privacy Enhanced Mail (PEM).

General Description of MD5

- After some initial processing, MD5 processes the input text in 512-bit blocks, divided into sixteen 32-bit sub-blocks. The output of the algorithm is a set of four 32-bit blocks, which concatenate to form a single 128-bit hash value.
- The main loop, which continues for as many 512-bit blocks as there are in the message, consists of four rounds of sixteen operations each. Each operation performs a nonlinear function on three of four 32 bit variables. The result is then added to the fourth variable.

Secure Hash Algorithm (SHA)

Secure Hash Algorithm (SHA)

- NIST, with assistance from NSA, designed the Secure Hash Algorithm (SHA) for use with the DSA. The standard is known as the Secure Hash Standard (SHS).
- When a message of any length is input, The SHA produces a 160-bit message digest.
- SHA is very similar in operation to MD4. It differs in that it adds an additional expansion operation, an extra round and the whole transformation was designed to accommodate the DSS block size for efficiency.
- Most cryptographers feel that the SHA is more secure than MD5 because of its fundamental design as well as its resistance to brute force attack on the 160-bit message digest versus the 128-bit digest produced by MD5.

Want to know more?

If you are interested in learning more about hash functions and other cryptographic techniques then you might want to read Chapter 14 of the following book:

*Applied Cryptography
Protocols, Algorithms, and Source Code in C*
Second Edition
by
Bruce Schneier

This page is intentionally blank.

Section 8

Cryptographic Protocols and Applications

Protocols

The Purpose of Protocols

- In daily life, there are informal protocols for almost everything that we do; ordering over the telephone, playing poker, exchanging cash for products or services, banking, voting in an election, to mention just a few. We don't give much thought to these protocols since they have evolved over time and everyone knows how to use them and how they work.
- Increasingly, people are communicating and doing business over computer networks rather than face-to-face as they have in the past. Many face-to-face protocols rely on people's presence to ensure fairness and security; however, in our new cyber world we do not have that luxury. It is therefore necessary to develop formal protocols for computing that can ensure that business is conducted fairly, honestly and securely.

Protocols

Protocol

An orderly sequence of steps taken by two or more parties to accomplish some task.

- Characteristics
 - 1. Established in advance
 - 2. Mutually subscribed
 - 3. Unambiguous
 - 4. Complete

Arbitrator Protocol

- Arbitrator
 - A Trustworthy, disinterested third party.
 - Directly involved in transaction.
 - A person, program or machine.
- Disadvantages
 - Suspicion
 - Cost
 - Delay
 - Bottleneck
 - Secrecy

Adjudicated Protocols

- Adjudicator
 - A third party who judges whether a transaction was conducted fairly.
 - A notary public
- Disadvantages
 - Detects failure to cooperate after the fact.

Self Enforcing Protocols

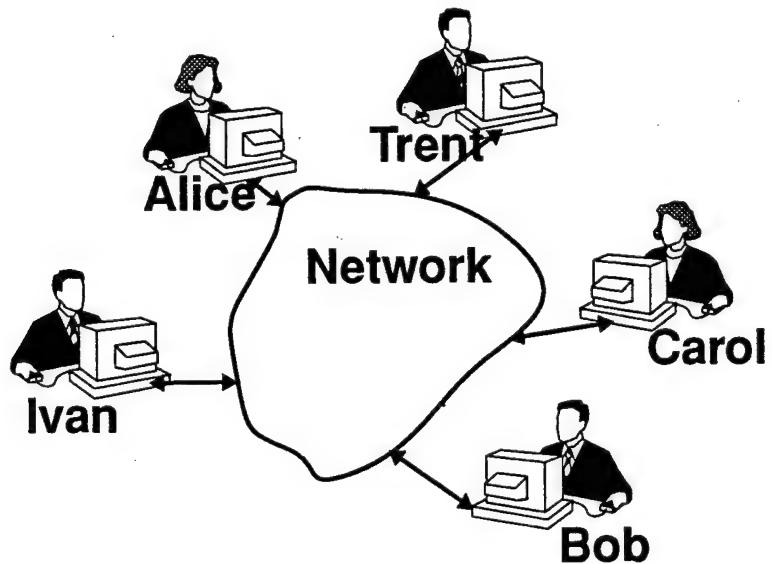
- Advantages
 - Guarantees fairness
 - No Outsider needed

Using DES To Support Secrecy

Alice wishes to send a secret message M to Bob.

Both Alice and Bob possess K_{AB}

1. Alice encrypts M with the DES key K_{AB} , producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with the DES key K_{AB} , producing M

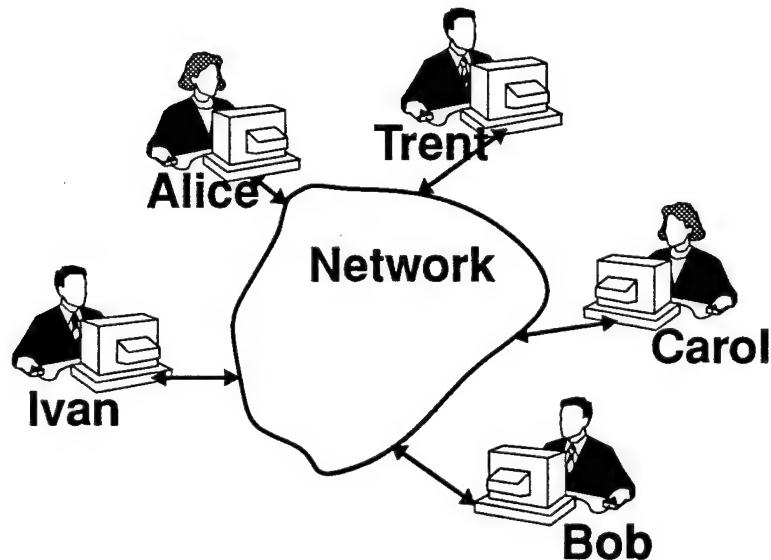


- Since both only Alice and Bob share the key K_{AB} , nobody else can read Alice's secret message M , unless they have managed to obtain a copy of the private key K_{AB} .

Using DES To Support Authenticity

Alice wishes to send a message M to Bob such that Bob is assured that the message could only have originated from Alice.

1. Alice encrypts M with the DES key K_{AB} , producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with the DES key K_{AB} , producing M

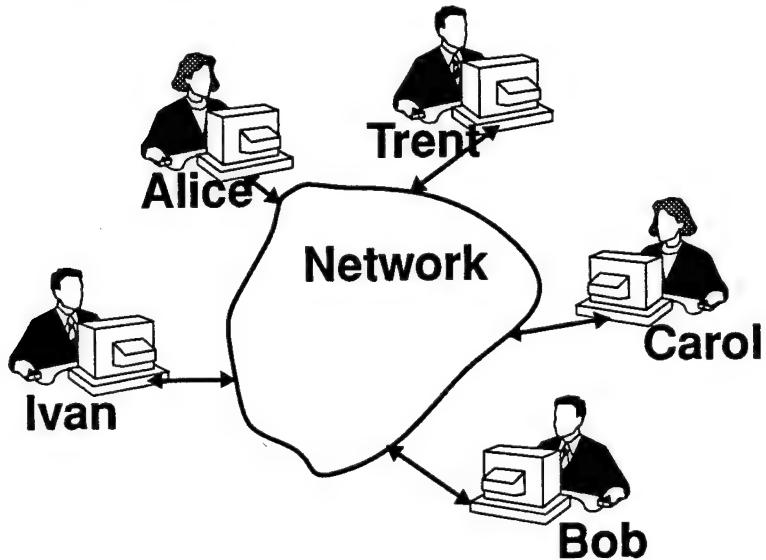


- Since both only Alice and Bob share the key K_{AB} , Bob knows that Alice is the only person who could have sent M. Nobody else could have sent it, unless they have managed to obtain a copy of the private key K_{AB} .

Using DES To Support Integrity

Alice wishes to send a message M to Bob such that Bob is assured that the message was not modified during its transit from Alice to Bob.

1. Alice encrypts M with the DES key K_{AB} , producing M'
2. Alice encrypts M with the DES key K_{AB} , using CBC mode and produces a 64 bit MAC
3. Alice transmits both M' and the 64 bit MAC to Bob
4. Bob decrypts M' with the DES key K_{AB} , producing M
5. Bob encrypts M with the DES key K_{AB} , using CBC mode and produces a 64 bit MAC'
6. Bob compares MAC and MAC' to see if they are the same.



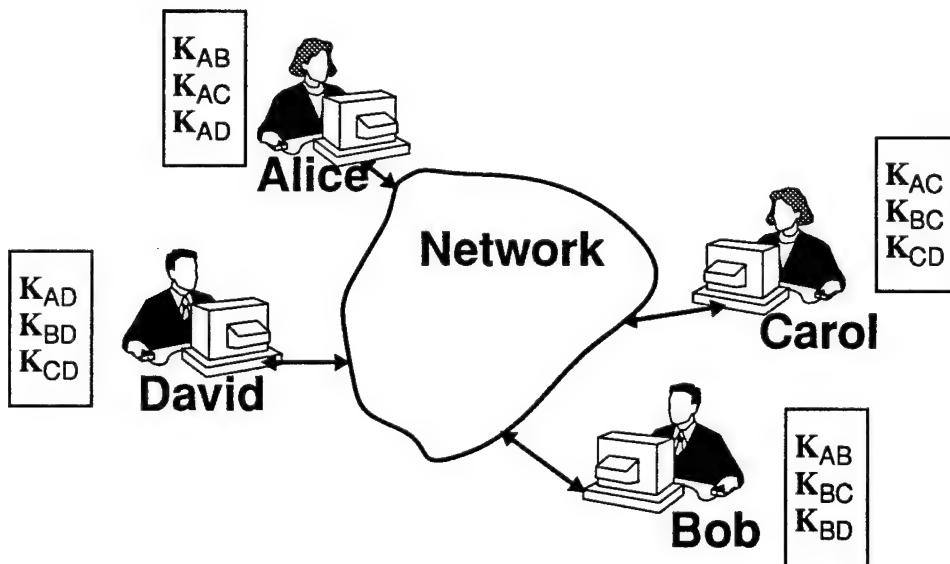
- Since both MAC and MAC' are the same Bob knows that the message M has not been altered.

Disadvantages of Conventional Key Systems

- With a conventional key system a separate key is needed for every pair of users.

$n * (n-1)/2$ keys are required for n users.

Example
3 users requires three keys k_{AB} , k_{AC} and k_{BC}
4 users requires six keys k_{AB} , k_{AC} , k_{BC} , k_{AD} , k_{BD} and k_{CD}
In general, we are choosing from n items k at a time or: $n * (n-1)/2$ keys are required



KEY MANAGEMENT IS A NIGHTMARE!!

Disadvantages of Conventional Key Systems

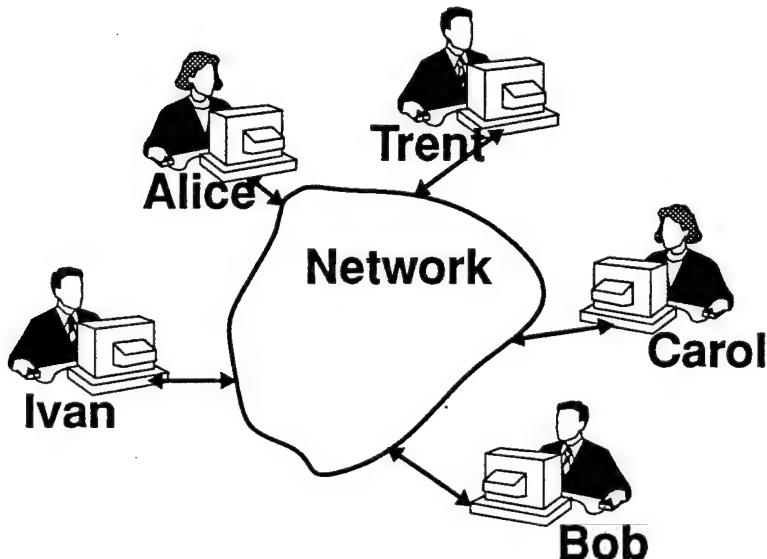
Distribution of Shared Keys:

- Keys must be distributed in a secure manner.
 - Bonded courier.
 - Registered mail.
- As previously mentioned, the amount of information a cryptanalyst can gain about a key is proportional to the number and length of messages encrypted with the key.
- Thus, for security reasons, keys should be changed periodically, which means that keys need to be securely distributed fairly often.

Using RSA To Support Secrecy

Alice wishes to send a secret message M to Bob.

1. Alice encrypts M with Bob's public key $K_{BOB-PUB}$, producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with his private key $K_{BOB-PRI}$, producing M



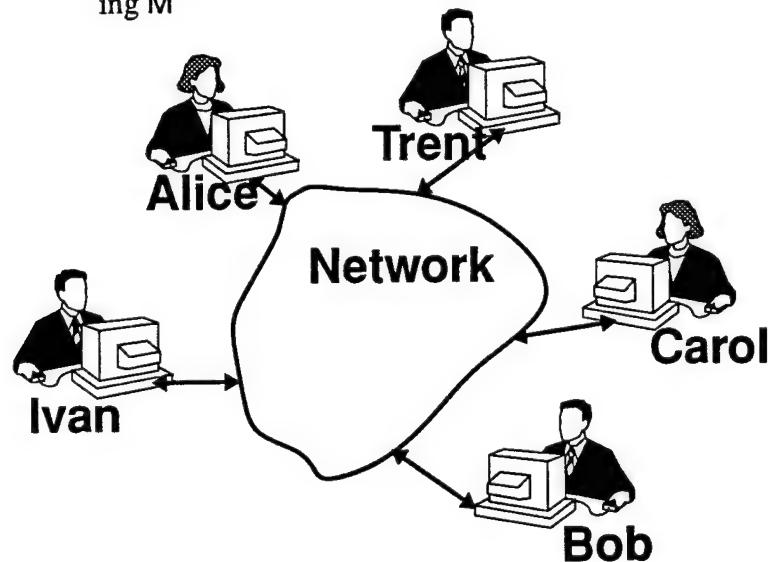
Problem!!

- Nobody else can read M since Bob is the only one who possesses $K_{BOB-PRI}$; however, Bob has no assurance that M came from Alice since anyone could have used his public key $K_{BOB-PUB}$.

Using RSA To Support Authenticity

Alice wishes to send a message M to Bob such that Bob is assured that the message could only have originated from Alice.

1. Alice encrypts M with Alice's private key $K_{ALICE-PRI}$, producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with Alice's public key $K_{ALICE-PUB}$, producing M



Problem!

- This protocol ensures authenticity, but secrecy is non-existent since anyone can obtain Alice's public key $K_{ALICE-PUB}$. Thus M is accessible to an eavesdropper like Ivan.

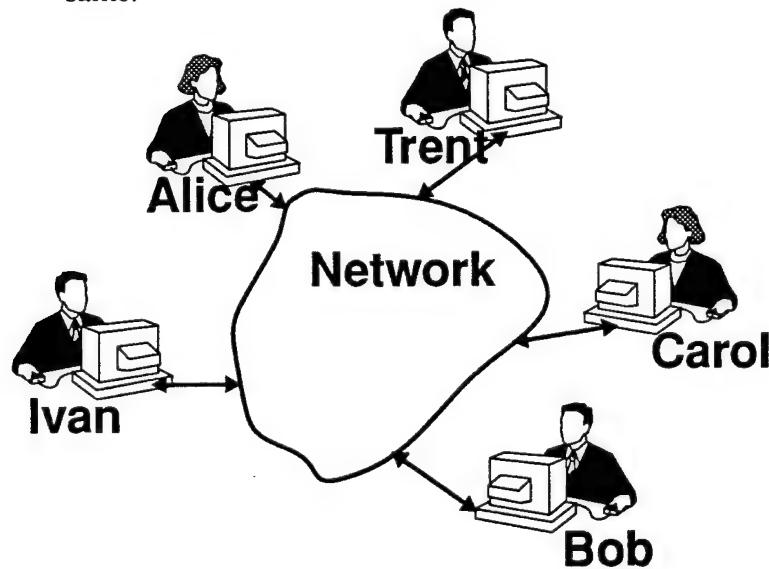
Problem!

- Bob is still not absolutely certain that M' was not altered in transit.

Using RSA To Support Secrecy, Authenticity and Integrity

Alice wishes to send a secret message M to Bob such that Bob is assured that the message could only have originated from Alice and that the message was not modified during its transit from Alice to Bob.

1. Alice uses a mutually available hash function H to produce a hash $H(M)_{ALICE}$ of the original message M .
2. Alice signs $H(M)_{ALICE}$ by encrypting it with her private key $K_{ALICE-PRI}$ producing $H(M)'_{ALICE}$
3. Alice encrypts M with Bob's public key $K_{BOB-PUB}$, producing M' .
4. Alice transmits both M' and $H(M)'_{ALICE}$ to Bob
5. Bob decrypts M' with his private key $K_{BOB-PRI}$, producing M
6. Bob uses the same hash function H to produce a hash $H(M)_{BOB}$ of the message M he just decrypted in step 5.
7. Bob decrypts $H(M)'_{ALICE}$ using $K_{ALICE-PUB}$.
8. Bob compares $H(M)_{ALICE}$ and $H(M)_{BOB}$ to see if they are the same.



Public-Key Systems Issues

Distribution of Keys:

To some degree, public key systems solve the key distribution problem that private key systems suffer from.

- The number of keys that a large community requires can be greatly reduced if the members of the community use a public key scheme instead of a private key scheme.
- If a community of n members uses a public key scheme they will only require n private keys and n public keys.
 - This is a total of $2n$ keys.
- There is no need for maintaining secrecy when distributing public keys.

Note that there is no need for secrecy, but there is a big need for authenticity.

- Consider the following scenario:
- Alice computes a private key / public key pair and sends the public key to Bob in a message such that Bob thinks that the public key is from Carol.
 - I.e., Bob thinks that this key is Carol's public key.
- Now if Bob wants to send a secret to Carol, he would encrypt the secret with Carol's public key and send it off to Carol.
- Alice intercepts the message and decrypts the message with the private key and reads the secret.
- This example illustrates the need for authenticity when receiving public keys.
 - This problem leads to public key management and certificate authority schemes.
- More on this topic later.

Efficiency Considerations

Public Key Versus Private Key Algorithms:

- Private key algorithms (such as DES and IDEA) are much faster than public key algorithms (such as RSA).

Question:

- How do we take advantage of public key cryptography for key distribution and private key cryptography for bulk encryption?

Answer:

- Use a hybrid scheme (such as PGP).

Hybrid Schemes:

- In a hybrid scheme, a public key algorithm is used to encrypt (and decrypt) a shared key (such as a DES key).
- The message is encrypted with the shared key.
- The encrypted message and the encrypted shared key are transmitted.
- The receiver, decrypts the shared key (using a public key algorithm) and then uses the shared key to decrypt the message.

Analysis of the hybrid scheme:

- Only keys (which are relatively short) are encrypted using the slow public key algorithm.
- The message (which may be very long) is encrypted with the fast shared key algorithm.
- The shared key may only be used for one message.

Digital Signatures

Digital signatures.

- A digital signature is the electronic analogue of a handwritten signature. A common feature is that they must provide the following:
 - A receiver must be able to validate the sender's signature.
 - A signature must not be forgeable.
- There are two major variants of implementation:
 - True signatures.
 - Arbitrated signatures.
- In a true signature system, signed messages are forwarded directly from signer to recipient. In an arbitrated system, a witness (human or automated) validates a signature and transmits the message on behalf of the sender. The use of an arbitrator may be helpful in event of key compromise as noted below.
- Digital signatures provide authentication, nonrepudiation and integrity checks. In some settings authentication is a major consideration; In some cases it is desirable even when secrecy is not a consideration.
- When using a public key scheme, encrypting a message (or the hash of a message) with a private key is effectively "signing" the message, since only one person in the world has the private key.

Key Management in Shared Key Systems

Conventional system key management.

- In a conventional (one-key) system, two users who wish to communicate securely must first securely establish a common key. One possibility is to employ a third party such as a courier. In practice, it may be necessary to establish a new key from time to time for security reasons. This may make use of a courier or similar scheme costly and inefficient.
- An alternative is for the two users to obtain a common key from a central issuing authority with whom each can communicate securely. Security is then a major consideration: a central authority having access to keys is vulnerable to penetration. Due to the concentration of trust, a single security breach would compromise the entire system. In particular, a central authority could engage in passive eavesdropping for a long period of time before the practice was discovered; even then it might be difficult to prove.
- In large networks it might become a bottleneck, since each pair of users needing a key must access a central node at least once. Additionally, failure of the central authority could disrupt the key distribution system. A hierarchical (tree-structured) system, with users at the leaves and key distribution centers at intermediate nodes may be one way to alleviate this problem. However, this creates a new security problem, since a multiplicity of entry points for intruders is created. Furthermore, it might be inefficient unless pairs of users communicating frequently were associated to a common subtree; otherwise the root of the tree would, once again, become a bottleneck.
- Some of these disadvantages can also be mitigated by a public-key approach to key distribution.

Key Management in Public Key Systems

Public-Key System Key Management.

- Prior to using a public-key cryptosystem for exchanging conventional secret keys, users Alice and Bob must exchange their public keys. It is a simpler problem than exchanging secret keys, since public keys do not require secrecy in storage or transit. Public keys can be managed by an on-line or off-line directory service; they can also be exchanged directly by users. However, authenticity is an issue. If Alice thinks that $K_{IVAN-PUB}$ is really $K_{BOB-PUB}$ then Alice might encrypt using $K_{IVAN-PUB}$ and inadvertently allow Ivan to decrypt using $K_{IVAN-PRI}$. A second problem is integrity: any error in transmission of a public key will render it useless. Hence some form of error detection is desirable. Regardless of the scheme chosen for public key distribution, at some point a central authority is likely to be involved. However, exchange of public keys between users need not involve the central authority, since the primary concern is with authenticity. Therefore, the implications of compromise of the central authority are not as severe as it would be in a conventional key system.
- Validity is an additional consideration: a user's public key may be invalidated because of compromise of the corresponding private key, or for some other reason such as expiration. This creates a stale-data problem in the event that public keys are stored or accessed through a directory.

Definition

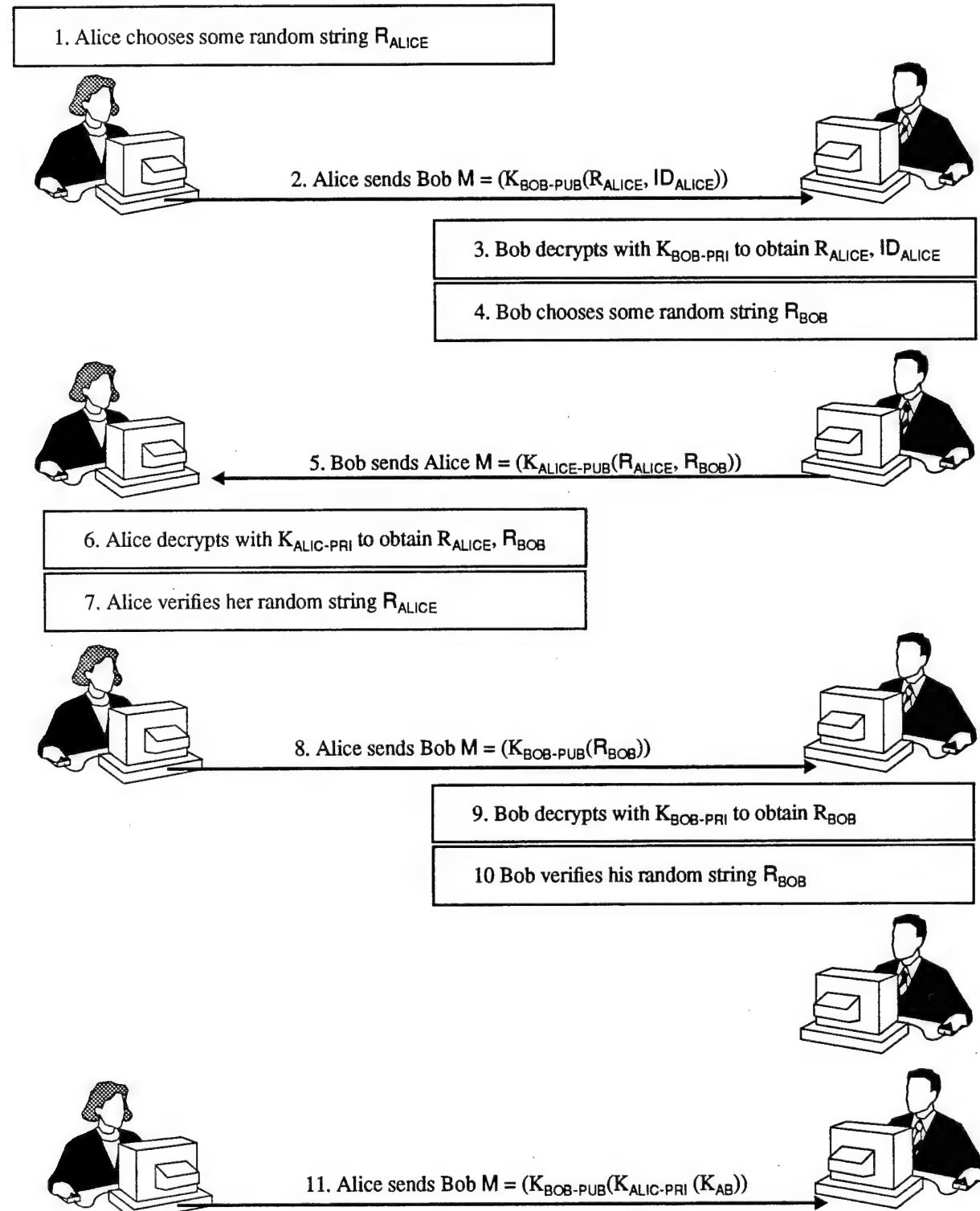
- A *nonce* is a random number that is often used in protocols to prevent the "replay" problem.

Example use:

- Alice generates a new nonce and sends a message containing the nonce. If she then receives a return message containing the same nonce value, she knows that the return message is not a replay of a previous message that had been sent to her.

Key Distribution Example

An authentication protocol in real time used for the exchange a DES key.



Key Management Via Certificates

Use of Certificates in Public Key Systems.

- A technique to obtain a solution to both authenticity and integrity in distribution of public keys is the use of certificates. A certificate-based system assumes a central issuing authority CA as in the secret-key case. Again it must be assumed that each user can communicate securely with the CA. This is relatively simple since it merely requires that each user possess K_{CA-PUB} , the public transformation of the CA. Then Alice may register $K_{ALICE-PUB}$ with the CA. Since $K_{ALICE-PUB}$ is public, this might be done via the postal service, an insecure electronic channel, a combination of these, etc.
- Normally Alice will follow some form of authentication procedure in registering with the CA. Alternatively, registration can be handled by a tree-structured system: the CA issues certificates to local representatives (e.g., of employing organizations), who then act as intermediaries in the process of registering users at lower levels of the hierarchy.
- In any case, in return Alice receives a certificate signed by the CA and containing $K_{ALICE-PUB}$. That is, the CA constructs a message M containing $K_{ALICE-PUB}$, identification information for A, a validity period, etc. Then the CA computes $CERT_{ALICE} = K_{CA-PRI}(M)$ which becomes Alice's certificate. $CERT_{ALICE}$ is then a public document which contains both $K_{ALICE-PUB}$ and authentication, since the certificate is signed by the CA. Certificates can be distributed by the CA, by users, or used in a hierarchical system. The inclusion of the validity period is a generalization of timestamping. The importance of timestamping is in guarding against the use of compromised keys.
- However, the problem of stale data is not wholly solved by timestamping: a certificate may be invalidated before its expiration date, because of compromise or administrative reasons. Hence if certificates are cached by users (as opposed to being redistributed by the CA each time they are used), the CA must periodically issue lists of invalidated certificates.

Key Management Via Certificates

A phone-book approach to certificates.

- Some of the features of the previous schemes could be combined in a phone-book approach, using an electronic equivalent such as a floppy disk containing certificates. This would optimize ease of use since a user could communicate securely with another by accessing the latter's certificate very rapidly. However, again the central authority would have to issue "hot lists". Periodic distribution of the compilations of certificates would be a separate management process. Additionally, the security of such a scheme is clearly in question since phone-book entries might contain errors, or entries could be altered.

Certificate	
Name:	Alice
Address:	alice@host.domain
Date Issued:	950901
Date Expires:	960831
Public Key:	clpg55kzxplvwqlfdrg tuoksfidolbkfgcsdarp 0qxsjhrxfjsdf2yun0ql dxklrtortwdgr6ee8l4
Signed: Mr. Trustworthy Trent	

Key Management Via Certificates

Decentralized Management.

- Users may be responsible for managing their own certificates. In this event the protocol is much simpler. When A wishes to initiate communication (e.g., exchange of a secret key) with B, he sends a message to B containing A's certificate, A's identification, and other information such as a date, random number etc. as described in the protocol in the previous section. This message also requests B's certificate. Upon completion of the certificate exchange, employing some protocol such as the handshake above, A and B will possess each other's authenticated certificates. A can validate the certificate C_B by decrypting the certificate with K_{CA-PUB} . Then K_{B-PUB} may be retrieved. The certificate must contain information properly identifying B to A, so that an intruder cannot masquerade as B. The certificate must also have a validity period. In turn B may proceed similarly.
- The central authority must periodically issue lists of certificates which have become invalid before their expiration dates due to key compromise or administrative reasons. It is likely that in a large system this would be done, e.g., on a monthly basis. Hence a user receiving a certificate from another user would not have complete assurance of its validity, even though it is signed by the CA. Thus this system trades higher efficiency for some security loss, compared to the previous scheme.
- For greater assurance of validity, a user could access a centrally-managed list of invalid certificates; presumably these would be very current. This would require on-line access to a central facility, which would create the same type of bottleneck we have noted previously. However, the accesses would be very quick, since presumably only a certificate sequence number would be transmitted.
- Another on-line possibility would be for the central authority to enforce coherence by a global search of cached certificates each time a certificate is invalidated. Again there is a trade-off of validity assurance and efficiency.

Certificate Example

Sample PGP certificate and signed message hash

```
- -----BEGIN PGP PUBLIC KEY BLOCK-----
```

Version: 2.6.i

```
mQCNAi+btRkAAEAEAKxQ9HwqfsQc9apOIQmFTo2wqbCL6Q1xlvN6CjxkBbtviaLq
EgmVPnb/FGD5wwxDMjCCJDwBFfLLRwASQAyyy5RjukkZx1Gn8qHzmoyIOVTFOIJI
TFDWyVjMSSvUKACDqXv/xVFunsPlPc7d6f4MwxD1kw2BBpoV7k64di/cua4BAAUR
tCRTc2ggZGldHJpYnV0aW9uIGtleSA8eWxvQGNzLmh1dC5maT6JAJUCBRAvm7Vv
qRnF8ZYfSjUBAW7pBACQ7G2pYSTkBM5aOK2udb/m/YAAZ/NIY2emSgEJfYrAysSY
0yfbhKGt0K59fGSotmSrCmOpq0tgTMm7IQjsUr5ez1Ra/0Dv7e3xoGQYJ8764X9w
popC+u9JuxLeGTtgWYwPUZIHFcQanZsIUmCDr36kvesx/2wXBf8+StghMbA3vw==
=aGik
```

```
- -----END PGP PUBLIC KEY BLOCK-----
```

```
-----BEGIN PGP SIGNATURE-----
```

Version: 2.6.i

```
iQCVAgUBMAPhQqkZxfGWH0o1AQHgngP/dbcRUFqJF549VvVOWgDtAxu/UoO6hnei
26/OpczgH6j8+6fZh8TV81yVAh95K6EhHsKo85j5hXTmKSG3xLn6fw26q1DPGHpQ
Sa4xQ4oL20qcvgOeaEi3gZxxTD5etzdl8eBNbe8vSlkk91yrsAiZL7h8St7UHGsA
N5WqXSMI8pg=
=tXr9
```

```
-----END PGP SIGNATURE-----
```

Key Distribution Example

Exchange of DES key using a one-way authentication protocol with certificates:

1. Alice chooses some random string R_{ALICE}

2. Alice constructs $M = K_{ALICE-PRI}(T_{ALICE}, R_{ALICE}, ID_{BOB}, K_{BOB-PUB}(K_{AB}))$



3. Alice sends Bob (CERT_{ALICE}, M)

4. Bob decrypts CERT_{ALICE} with K_{CA-PUB} to obtain $K_{ALICE-PUB}$

5. Bob decrypts M with $K_{ALICE-PUB}$ to obtain $(T_{ALICE}, R_{ALICE}, ID_{BOB})$

6. Bob verifies $T_{ALICE}, R_{ALICE}, ID_{BOB}$

7. Bob checks R_{ALICE} to make sure that M is not a replay.



CLIPPER

The CLIPPER Chip

- The CLIPPER chip is one implementation of the SKIPJACK algorithm. The Clipper chip designed for the AT&T commercial secure voice products has the following characteristics.
 - Functions specified by NSA; logic designed by MYKOTRONX; chip fabricated by VLSI, Inc.
 - Resistant to reverse engineering against the most sophisticated techniques which may be used by a well funded adversary.
 - 15-20 MB/S encryption/decryption rate once synchronization is established.
 - The chip programming equipment writes (once) the following information into a special memory on the chip.
 - serial number (unique) unit key (unique) family key specialized control software
- 1. Upon generation (or entry) of a session key in the chip, the chip performs the following actions.
- 2. Encrypts the 80-bit session key under the unit key producing an 80-bit intermediate result.
- 3. Concatenates the 80-bit result with the 25-bit serial number and a 23-bit authentication pattern (a total of 128 bits).
- 4. Enciphers this 128 bits with the family key to produce a 128-bit cipher block chain called the Law Enforcement Field (LEF).
- 5. Transmits the LEF at least once to the intended receiving Clipper chip.
- 6. If law enforcement agencies want to decrypt the session, they intercept the LEF and messages of this session. They use the LEF to obtain the session key from the Escrow authorities.
- 7. The two chips use this field together with a random initialization vector (IV) to establish cryptographic synchronization.
- 8. Once synchronized, the CLIPPER chip uses the session key to encrypt/decrypt data in both directions.
- The chips can be programmed to not enter secure mode if the LEF has been tampered with (e.g. modified, superencrypted, replaced).
- CLIPPER chip prices currently cost \$16 unprogrammed and \$26 programmed.

CLIPPER

The Great Debate - Key Escrow

- The CLIPPER chip was intended to protect private communications while at the same time permitting government agents to obtain the keys upon presentation of legal authorization. The two halves of the unique unit key are to be held by two separate escrow agents and would allow the government to access the encrypted private communications. The use of CLIPPER, so far, is voluntary; however, government agencies are being strongly encouraged to adopt its use.
- The National Institute of Standards and Technology (NIST) and the Department of the treasury were designated on February 4, 1994 as the two escrow agents that will hold the keys in escrow.
- The topic of key escrow has been the subject of a great deal of heated debate over the past two years. On one side are those that feel that individual privacies are at risk and those that argue that law enforcement officials must be given the technological ability to do their jobs. The most notable proponent for key escrow is Dorothy Denning who believes that CLIPPER is necessary to stop crime. The Electronic Freedom Foundation (and notably John Perry Barlow) adamantly opposes the key escrow concept because the temptation for government to usurp the rights of private individual's rights is too tempting.

Anonymous Key Distribution

Anonymous Key Distribution

- Consider the problem of key distribution. If we assume that people do not have the ability to generate their own keys then they must use the services of a Key Distribution Center (KDC). The problem is that the keys must be distributed in such a fashion that no one may determine who gets what keys.

1. Alice generates a public/private key pair (for this protocol, she keeps both keys secret).
2. The KDC generates a continuous stream of keys.
3. The KDC encrypts the keys, one-by-one, with its own private key.
4. The KDC transmits the encrypted keys, one-by-one, onto the network.
5. Alice chooses a key at random.
6. Alice encrypts the chosen key with her public key.
7. Alice waits for a while and sends the double encrypted key back to the KDC.
8. The KDC decrypts the double encrypted key with its private key, leaving a key encrypted with Alice's public key.
9. The KDC sends the encrypted key back to Alice.
10. Alice decrypts the key with her private key.

Nonrepudiation

Nonrepudiation.

- Nonrepudiation is contingent upon users keeping their private keys secret. If Alice's private key $K_{ALICE-PRI}$ should be compromised, then Alice might be able to repudiate messages sent even before the compromise.
- The use of a central authority is suggested for this purpose. In this scheme, the receiver of a message sends a copy to the central authority. The latter can attest to the instantaneous validity of the sender's signature (i.e., that it has not been reported that the sender's private key has been compromised at the time of sending).
- In a public-key system augmented by a hash function H , Alice might send a message M to Bob as follows (ignore secrecy considerations): Alice sends M and a signed hash $H(M)'_{ALICE}$ to Bob. Bob uses Alice's public key $K_{ALICE-PUB}$ to retrieve $H(M)$, then computes $H(M)_{BOB}$ and compares the two values for authentication. For nonrepudiation, Bob retains M , $K_{ALICE-PUB}$ and $H(M)'_{ALICE}$. If Alice attempts to repudiate M , a judge can use the three items to resolve the dispute by completing the same steps Bob did and attesting to the validity of Alice's private key at the time of transmission of M .
- The preceding schemes satisfy another desirable property: in the adjudication process, they do not compromise security by exposing private keys to a judge.
- Another solution involves timestamps. This again may involve a network of automated arbitrators, but is very simple in nature. Receivers obtain timestamps along with the received message. If a receiver needs to be sure of the validity of a signature, he may check the validity of the sender's private key by checking with a central authority. As long as the received message is timestamped before the validity check, the receiver is assured of nonrepudiation. If users are permitted to change their keys, a central authority should retain past keys for use in disputes which may arise later.

Secret Sharing Algorithms

Secret Sharing Algorithms

- Let's assume that you're setting up a launch program for a nuclear missile. You want to make sure that no single individual can initiate a launch. You have five officers whom you feel can be trusted with the individual secret codes to initiate a launch; however, you additionally desire that only three officers need be present to launch a missile.
- This problem can be solved by a secret sharing scheme, called a threshold scheme. In its simplest form any message (a launch code in this case) can be divided into n pieces, called shadows, such that any m of them can be used to reconstruct the entire message.
- Lets say you desired to create a (3,5) threshold scheme to satisfy our launch code scheme, then we could use the following technique developed by Shamir.
 - Generate a quadratic polynomial $ax^2 + bx + M \pmod{p}$ in which p is a random number. The coefficients, a and b , are chosen at random and are kept secret and discarded after the shadows are handed out. M is our secret launch code. The prime p is made public.
 - The shadows k_i are obtained by evaluating the polynomial at five points $k_i = F(x_i)$.
 - Since the quadratic polynomial has three unknown coefficients, a , b , M , any three shadows can be used to create three equations since the other two equations are redundant.

Secret Sharing Algorithms

Secret Sharing Algorithms

Example

Let:

$$\begin{aligned} M &= 11 && \text{(our secret launch code)} \\ a &= 7 & b &= 8 && \text{(our chosen randoms)} \end{aligned}$$

Which generates the quadratic:

$$F(x) = 7x^2 + 8x + 11 \pmod{13}$$

We create the following shadows:

$$\begin{aligned} k_1 &= F(1) = 7 + 8 + 11 = 0 \pmod{13} \\ k_2 &= F(2) = 28 + 16 + 11 = 3 \pmod{13} \\ k_3 &= F(3) = 63 + 24 + 11 = 7 \pmod{13} \\ k_4 &= F(4) = 112 + 32 + 11 = 12 \pmod{13} \\ k_5 &= F(5) = 175 + 40 + 11 = 5 \pmod{13} \end{aligned}$$

To reconstruct M from three of the shadows, let's say, k_2 , k_3 and k_5 we solve the following set of linear equations to find M :

$$\begin{aligned} 3 &= a * 2^2 + b * 2 + M \pmod{13} \\ 7 &= a * 3^2 + b * 3 + M \pmod{13} \\ 5 &= a * 5^2 + b * 5 + M \pmod{13} \end{aligned}$$

Blind signatures

Blind signatures

- Usually we desire people to be aware of the contents of a document before signing it; however, there are times when we wish to have people sign a document without their seeing the contents. This has an obvious application in the real world, specifically to the notarization process. Less obvious perhaps is that we can use blind signatures in voting protocols.
- Let's assume that Bob is a notary public. Alice wants him to sign a document but does not want him to have any idea what he is signing. Bob doesn't care because he is just certifying that he notarized the document at a given time. The following simple protocol can be used.
 1. Alice takes the document and multiplies it by a random value, called a blinding factor.
 2. Alice sends the blinded document to Bob.
 3. Bob signs the document and returns it to Alice.
 4. Alice divides out the blinding factor, leaving the original document signed by Bob.

Secure Elections

Secure Elections

- Although on the surface it may seem like a very simple protocol to develop, secure voting can involve rather detailed and complicated protocols. We will look at only a few of the more simplistic protocols which involve a Central Tabulating Facility.
- Ideally, the protocol we desire has the following properties:
 1. Only authorized voters can vote.
 2. No one can vote more than once.
 3. No one can determine for whom anyone voted.
 4. No one can change anyone else's vote without being discovered.
 5. All voters can make sure that their vote has been taken into account in the final tabulation.

Simplistic Secure Voting Protocols

Simplistic Secure Voting Protocols

- Let's look at a very simplistic protocol that does not work:
 1. All voters encrypt their vote with the public key of the CTF.
 2. All voters send their vote to the CTF.
 3. The CTF decrypts the votes, tabulates them and makes the result public.

O-o-o-ps!

This protocol does not work because it violates every one of the characteristics we desire in a voting protocol.

- Let's try an improvement over the first protocol that satisfies some of our properties but still falls short of the mark:
 1. All voters sign their vote with their private key.
 2. All voters encrypt their signed vote with the CTF's public key.
 3. All voters send their vote to the CTF.
 4. The CTF decrypts the vote, checks the signatures, tabulates the results, and makes the results public.

O-o-o-ps!

This protocol is only a slight improvement since it only satisfies properties (1) and (2) of our ideal protocol.

Voting with Blind Signatures

Voting with Blind Signatures

- The problem that we now face is how to dissociate the vote from the voter, while still maintaining authentication:
 1. All voters generate a pair of votes (“yes” and “no”) to which a very large randomly generated serial number is attached.
 2. All voters blind their pairs of votes and send them to the CTF.
 3. The CTF checks its database to make sure voters have not submitted blinded votes for signature previously. Then it signs the votes and sends them back to the voters and stores the names of the voters in its database.
 4. The voters unblind the messages, leaving a “yes” and “no” vote, each of which has been signed by the CTF.
 5. The voters choose one of the votes and encrypts it with the CTF’s public key.
 6. The voters send in their votes.
 7. The CTF decrypts the votes, checks the signature, checks its database for a duplicate serial number, saves the serial number, and tabulates the votes. It publishes the results of the election, along with every serial number and its associated vote.

A-a-b-a-h!

- The protocol above works even better if all votes are collected in an electronic ballot box prior to giving it to the CTF, since this would make it impossible for the CTF to keep track of who sent in what vote.
- The problem with this protocol is that it does involve a CTF which must be trusted. In order to eliminate the CTF we must resort to rather complex and cumbersome protocols which appear to be extremely impractical for large scale elections.

Digital Cash

Digital Cash

- Once again an everyday protocol which we frequently take for granted, and which also appears to have a relatively simple solution in cyber-space, requires a complex and cumbersome solution. There is no single solution which fits our concept of an ideal digital cash world. The following characteristics have been identified as six properties of an ideal digital cash system:
 1. Independence - The security of the digital cash is not dependent on any physical location. The cash can be transferred through computer networks.
 2. Security - The digital cash cannot be copied or reused.
 3. Untraceability - No one can trace the relationship between users and their purchases.
 4. Off-Line Payment - The point of purchase does not need to be linked to a host to process the user's payment.
 5. Transferability - The digital cash can be transferred to other users.
 6. Divisibility - A piece of digital cash in a given amount can be subdivided into smaller pieces of cash in smaller amounts.
- There are a number of digital cash systems which have been proposed and several implementations currently exist; however, no practical solution has been implemented which satisfies all of the constraints above.
- Okamoto and Ohta have developed a schema which satisfies all the constraints. The total data transfer for a payment is about 20 kilobytes, and the protocol can be completed in several seconds.

S/Key - One-Time Password System

S/Key - One-Time Password System

- Developed and implemented at Bellcore.
- One-time password system, to counter eavesdropping on network connections to obtain user/account information and passwords.
- The user's secret password never crosses the network during login.
- User's secret password is never stored anywhere, including on the host being protected.
- Based on publicly available hash function algorithm (MD4/MD5).
- Takes 8 bytes of input to MD4 then by folding the output byte pairs in the 16-byte MD4 to produce an 8-byte output (yielding a 64-bit password).

How one-time passwords are generated:

1. The very first one-way password is created by running the secret password s through the hash function f some specified number of times, N .

$$p_0 = f^N(s)$$

2. The next one-way password is generated by running the secret password through the hash function $N-1$ times;

$$p_1 = f^{N-1}(s)$$

3. In general, each subsequent one-time password p_i is generated by:

$$p_i = f^{N-i}(s)$$

An eavesdropper will not be able to generate the next one-time password in the sequence because doing so would require inverting the hash function.

S/Key - One-Time Password System

How one-time passwords are generated:

1. The host is initially given p_0 .
2. When a client attempts to be authenticated, the seed and the current value of i are passed to the client.
3. The client returns the next one-time password by taking a the seed value, which is concatenated to the password, and running the modified secret password through the hash function $i-1$ times.
4. The host temporarily saves the clients result, then applies the hash function to it.
5. If the result compares to its previously stored value then the temporary copy replaces the previously stored value, and the client is given access.

- After the user has used $N-1$ passwords then it is necessary to reinitialize the system through the use of *keyinit*, which is a special version of the UNIX *passwd* command.
- S/Key is available for anonymous *ftp* from: *thumper.bellcore.com* in the *pub/nmh* subdirectory

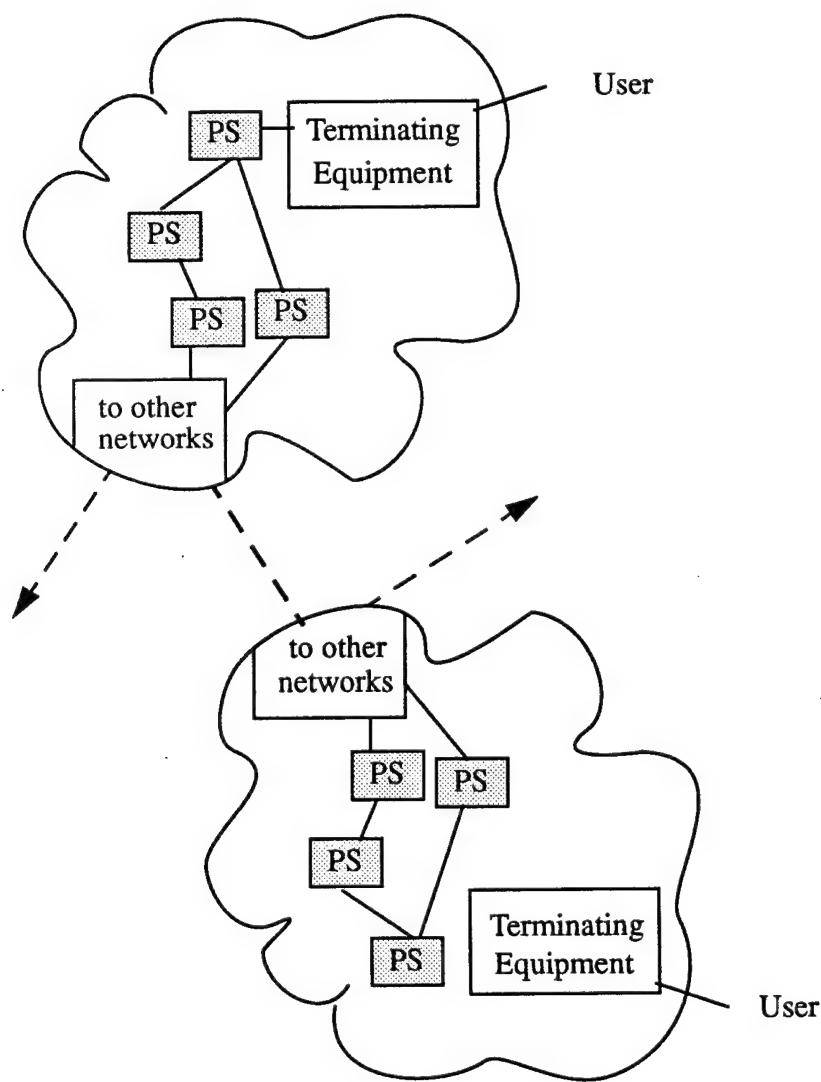
Section 9

Network Security

Introduction to Networks

Network

A collection of interconnected functional units providing data communications services among components attached to it. These components are comprised of both hardware and software



Threats to Network Security

Their distributed nature and the possibility of increased interconnectivity render networks more vulnerable than monolithic systems.

Traffic Flow Analysis

Inference of information through the examination of message attributes rather than message contents

- Traffic frequency
- Source addresses
- Destination addresses
- Dominoes Pizza Traffic

Denial of Service

The prevention of authorized access to physical resources through theft or disruption. Delaying or denying time-critical access is a form of denial of service

- flooding of network with traffic
- blocking of transmissions based on addresses
- message replay

Spooing (Impersonation)

Use of services under a false identity. May obtain unauthorized access to information or may maliciously modify information

- replay of passwords
- modification of source addresses
- compromise of passwords
- message replay

Eavesdropping

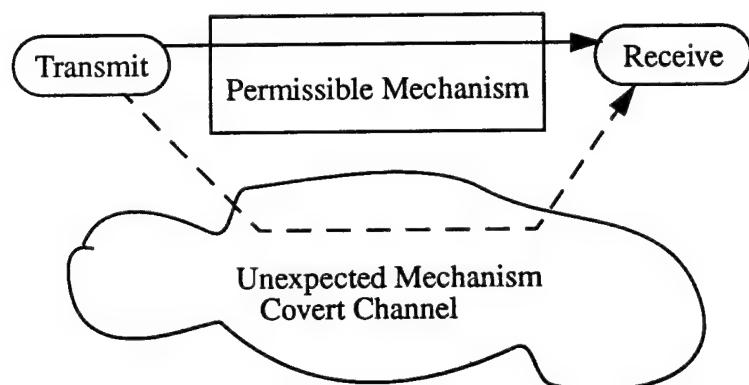
Illicit capture of information while it is enroute between communicating parties

- Radio link interceptions
- Wiretapping
- Emanations from communications equipment
- Grabbing unprotected ethernet packets in a LAN

Threats to Network Security

Covert Channels

The use of system mechanisms in an unexpected manner which causes the leakage of information in violation of the system security policy.



- Encodings using message length
- Encodings using message addresses

Network Security Services

There is overlap between these areas. Note that ISO lists the following: Non-repudiation, access control, authentication, data confidentiality, and data integrity

Access control

Enforcement of security policy when requests for access are made

Information Confidentiality

Protection of information from unauthorized disclosure

Information Integrity

Protection of information from unauthorized modification

Authentication and Non Repudiation

Insure the proper authentication of active system entities. Prevents impersonation or masquerading

Prevent the repudiation of prior events

- Proof of origin
- Proof of receipt

Availability

Insure that the network services are both available and of appropriate quality

Confidentiality

Objective:

- Restrict access to data in transit or in storage to TCB components

Method:

- transform data to render contents unreadable -- encryption
- store data in system-only domain -- secured computing systems or trusted processor states
- Use of label-based controls, e.g. derived from Bell and LaPadula Model
 - Prohibit flow of information down in levels of confidentiality
 - Permit entities at higher confidentiality levels to observe information at lower confidentiality levels

Considerations for location of confidentiality controls

- Which protocol layer?
- Multiple protocol layers?
- End system provides?
- Intermediate system provides?

Problems

Key Management

- Key distribution centers for private keys
- Certificate servers for public keys

Separation of Encrypted from Unencrypted

- Where is the TCB?

Integrity

Objective:

- Prevent unauthorized modification of data. Includes
 - integrity of information in transmission packets
 - ordering of transmission packets
 - insuring transmission of complete message to intended destination

Method:

- use message authentication codes (MAC), also known as integrity locks
- Use of label-based controls, e.g. derived from Biba Integrity Model
 - Prohibit flow of information up in levels of integrity
 - Permit entities at lower integrity levels to observe information of higher integrity

Considerations for location of integrity controls

- Which protocol layer?
- Multiple protocol layers?
- End system provides?
- Intermediate system provides?

Problems

Management

- Protection of integrity enforcement mechanisms
- Determination of specific integrity attributes, e.g. labels or MACs

Authentication and Non-repudiation Services

Objective:

- Establish message origin (author)
- Obtain proof that message was sent.

Method:

- use signatures, time-stamps

Considerations for location of authentication and non-repudiation services

- Which protocol layer?
- Multiple protocol layers?
- End system provides?
- Intermediate system provides?

Problems

Protocol designs

- The protocols can be complicated.

Availability Services

Although desirable, cannot be specified in the same precise, global, and persistent manner as confidentiality and integrity policies

- Subjective
 - What is progress?
 - What is termination?
 - What is success?
 - what is sufficient?
 - Cannot look at individual components, entire system must be examined
- Authorized users may compete for system resources
- To reduce scope of problem, external users may need to be excluded
 - totally
 - provided with circumscribed services
- How are availability attributes defined?
- How are availability attributes allocated and revoked?

Network Security Mechanisms

Table 0: Overview of Network Security Mechanisms

	Encryption	Trusted Network Base	Physical Protection
Authentication	Yes. Validate authenticity of requestor of service. Sometime "certificates are produced for future requests	Yes. Mandatory integrity fields, but information must be under TNB control	Yes, should use error detection techniques to support physical protection
Access Control	No, but can help enforce TCB access control decisions	Yes, Label-based access control decisions	Restrict use of end system only to authorized users
Confidentiality	Yes, Semantics of data altered so that they are unintelligible	Yes, but data must remain under continuous control of the TNB, otherwise supplemental mechanisms, such as encryption, are needed	Separate users who should have access to information from those who should not
Integrity	Yes, Encrypted data may have internal checksum Cleartext may have cryptographically computed checksum	No in the case of communications Enforcement of Biba integrity model within system. Encryption for communications integrity	Malicious modification can be avoided by combined physical and personnel controls

Cryptography for Secure Networking

Objective:

transform information so that it is unreadable without use of a key to transform ciphertext back to plaintext

Permits secure communication over an insecure channel

Two kinds of Cryptography

Private Key -- Symmetric Key

- examples include Skipjack, DES
- Same Key and algorithm are used to encrypt and decrypt message

$$D_k(E_k(P)) = P$$

- Supports multiple receivers
- Key must be agreed upon by sender and receiver
 - trusted protocol for key distribution
 - synchronization of rekeying
 - deliver keys by trusted courier
- encipherment is in fixed block sizes
- high bandwidths supported
 - hardware implementations
 - stream mode -- 10^8 bps
 - change keys with each protocol data unit -- 10^6 to 10^7 bps

Public Key -- Asymmetric Key

Examples include RSA

Sender and receiver have different keys for encryption and decryption

$$D_{priv}(E_{pub}(P)) = P$$

The encryption key is public and may be stored in a “directory”

The decryption key must be protected

Bandwidth lower $\sim 10^6$ bps

Benefits of Cryptography

Cryptography contributes to communications security (COMSEC)

Confidentiality

- Once a sensitive message has been encrypted using a strong encryption algorithm, it can only be decrypted using the key
- Encrypted data is considered to be unclassified

Integrity

- Digital checksums
- Message authentication codes
- Integrity Lock, also known as “Spray Paint”
- Message Digests
 - likelihood of two messages producing the same message digest is low
 - no keys are required
 - MD5 is a popular choice

Authentication

- Use key to validate identity
- Private Key Authentication
 - Three-way protocol usual
 - Trusted key server required
 - Kerberos is an example
- Public Key Authentication -- digital signatures
 - NIST digital signature standard (DSS) - no privacy
 - Basic mechanism: sender signs message through encryption with private key and receiver authenticates by decrypting with public key
 - Replay is prevented by using time stamps
 - Digests can improve efficiency
 - Problem: public keys must be distributed

Access Control -- NO!

Cryptography does not provide access control. If a TCB selects pairwise keys based on security labels, then cryptography can support TCB access control decisions.

Challenges to Network Cryptography

Key Management

Classification of key is the least upper bound of the classifications of data encrypted with the key

Who should receive individual keys?

- each message
- each security level
- individual users
- groups of users
- hosts
- each connection

Key Generation

- need to have good random number generators
- need to avoid weak keys
 - all 1s
 - all 0's
 - etc.

Need Key Distribution and Revocation Plan

- may differ for symmetric and public keys
- need key revocation lists
- may choose to use mixed schemes
- public key cryptography to distribute symmetric (shared) keys
- symmetric key cryptography to encrypt messages
- Secure Data Network System (SDNS) supports key distribution relying on public key methods
- Blacker and Kerberos are systems where a particular network node is allocated key distribution responsibility

ISO Reference Model

Open Systems Interconnection (OSI)

- Describes computer network communications.
- Model describes peer-to-peer correspondence, relationship between corresponding layers of sender and receiver.
- Each layer represents a different activity performed in the actual transmission of a message.
- Each layer serves a separate function
- Equivalent layers perform similar functions for sender and receiver

Layer	Responsibility	Actions
Layer 7 Application	User Program	Initiates message; optional encryption
Layer 6 Presentation	System Utilities	Breaks message into blocks, text compression; optional encryption
Layer 5 Session	Operating System	Establishes user-to-user session, header added to show sender, receiver and sequencing information, recovery
Layer 4 Transport	Transport Manager	Flow control, priority service, information added concerning the logical connection
Layer 3 Network	Network Manager	Routing, message blocking into packets, routing information added to blocks
Layer 2 Data Link	Hardware	Transmission error recovery, message separation into frames; optional encryption, header and trailer added for correct sequencing and error detection
Layer 1 Physical	Hardware	Physical signal transmission, by individual bits

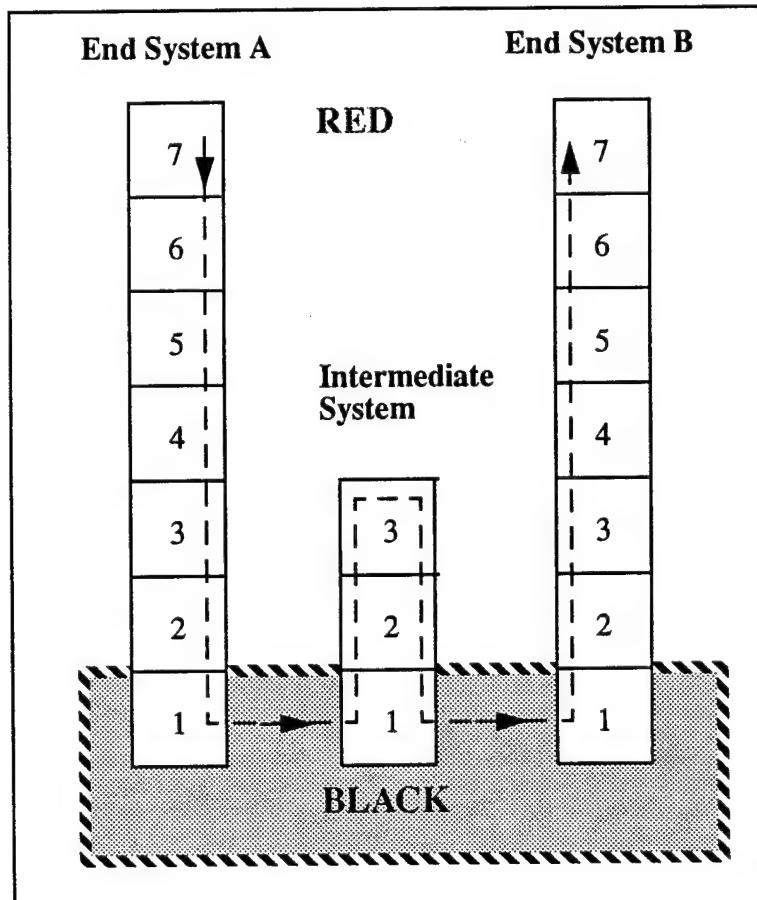
OSI Security Service Matrix

Services	1	2	3	4	5	6	7
peer entity authentication			X	X			X
data origin authentication			X	X			X
access control			X	X			X
connection confidentiality	X	X	X	X		X	X
connectionless confidentiality		X	X	X		X	X
selective field confidentiality							X
traffic flow security	X		X				X
connection integrity			X	X			X
connectionless integrity			X	X			X
nonrepudiation							X
	P h y s i c a l	D a t a L i n k	N e t w o r k	T r a n s p o r t	S e s s i o n	P r e s e n t a t i o n	A p p l i c a t i o n

Cryptography Placement Issues

- Data in Red is in the clear.
- Data in the Black is encrypted.
- Encryption takes place as data passes from the Red region to the Black region.
- Decryption takes place as data passes from the Black region to the Red region.

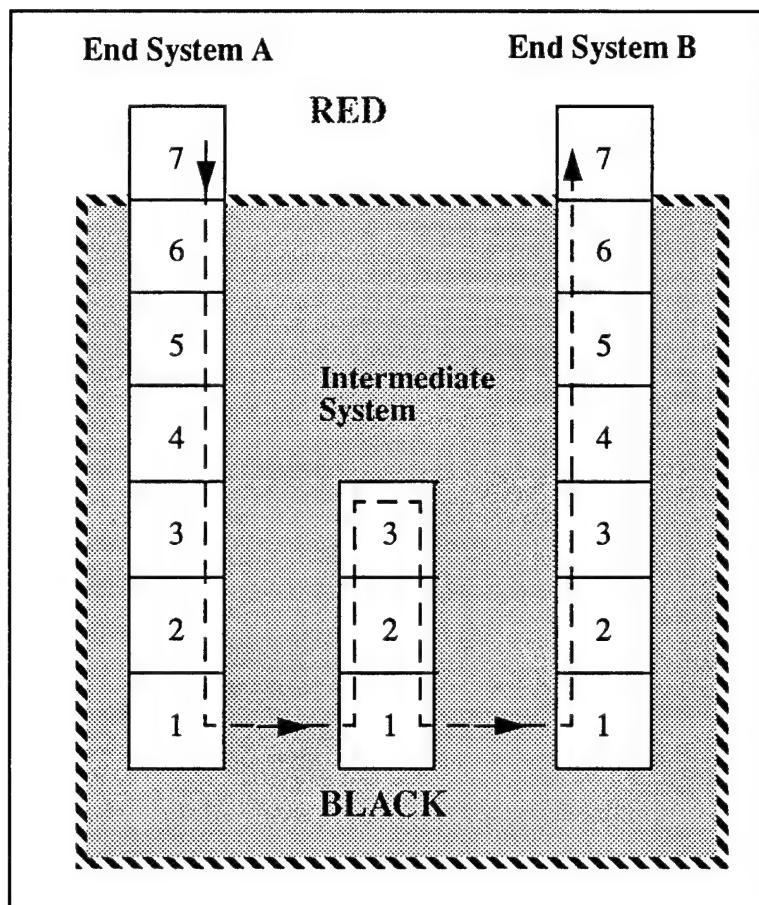
- Link Encryption -- Encryption is a link layer
- All Intermediate systems must be trusted
- Protects against compromise during transmission



Cryptography Placement Issues

End-to-End Encryption

- Encryption is at application layer
- Application information is protected
- Potential for many attacks at lower protocol levels



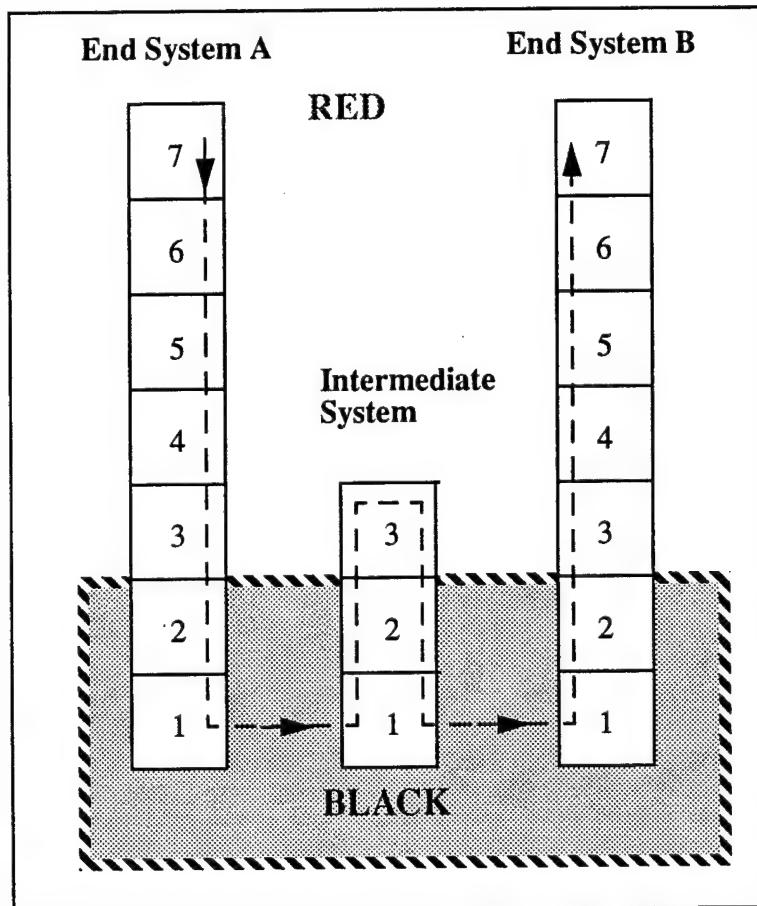
Cryptography Placement -- SILS

Standard for Interoperable LAN Security

IEEE 802.10 LAN Security Working Group

- vendors, government, and users

Security Services at layer 2 of OSI framework



Benefits and Disadvantages of SILS

SILS Permits Routing

Recall Layer 3 provides routing services

Advantages

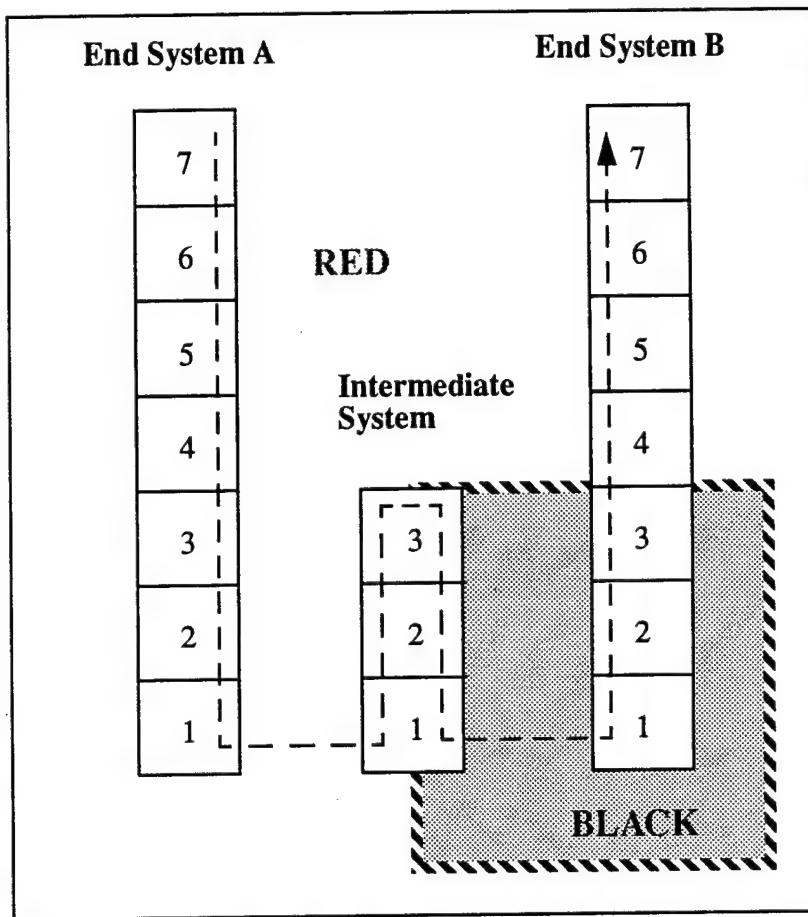
- routing may be optimized
- Many protocols exist at Layer 3, e.g. Novell, TCP/IP, etc.
 - With encryption at Layer 2, all are supported

Disadvantages

- with encryption below the routing layer, then clear text must be available in intermediate systems
- the addresses provided by the source will not necessarily be used because the router may choose “better” addresses
- all routing paths used must be able to encrypt/decrypt so that destination can decrypt and reconstruct the complete message
- Performance at high bandwidth may require multiplexing of multiple cipher streams

Mixed Systems

Here some systems use encryption while others do not



Challenges

- maintain DAC of End Systems
- separate messages according to security level
- keep encrypted and unencrypted information separate

Mixed Systems - continued

Red systems may be LANS

- inexpensive
- local protection
- use secure facilities
- do not need encryption, so performance is better
- must be trusted
- separate information by security level
- enforce DAC policy, which may include a policy regarding how messages will be routed when they are transmitted from the LAN

Intermediate systems must be trusted

- must separate Red and Black
- must protect keys and cryptographic methods
- may need to select cryptography based upon message security levels
- routing policy may need to be enforced

Black systems require no mandatory trust

- all messages are treated as having same level
- DAC routing policy may need to be enforced
- may choose to protect communications lines against
 - traffic analysis
 - denial of service

Where are Encryption Mechanisms Located?

Within the host system

- If you want to use encryption at the application level
- May use different encryption for different applications
 - files
 - e-mail
 - etc.

need trusted system

- separate Red/Black
- manage keys and cryptography
- information is read from Red and written to Black

As a front end device

- helps to localize encryption
- could be device on I/O port
- could be card on system bus (part of hardware, but not "in" the host system)

Network Evaluations

Introduction

The Trusted Network Interpretation (TNI) assumes that it is possible to evaluate a network under the TCSEC.

- implies that the network can be treated as if it were a monolithic computer system
- Strategy is to partition the TCB both logically and physically among components of the network.
- This results in a Network TCB (NTCB).

Evaluation procedure

- must have decomposition of overall network security policy into policies for individual components
- evaluate individual components
- use the network security architecture to support the assertion that the network is a sound composition of its components
- given a sound composition assert that since each supports its allocated policy correctly that overall network security policy is supported We will inspect the rationale for doing this.

Network Security Policies

Need to examine policy and how it will be allocated to Network components.

Security Policy

Security policy is broadly expressed in terms of people and information

- need a **single** uniform network security policy
- if there are multiple organizations involved, then the security policy needs to be defined during the **early** stages of the network development process

Mandatory Access Control Policy

- based on a comparison of labels associated with information with clearances of users.
- may need to merge systems of classes and clearances
- commercial organizations may have trivial MAC policies
- label-based, i.e., labels on data units and communications entities

Discretionary Access Control Policy

- these are much more diverse
 - modes of access
 - composition of groups
 - kinds of named objects for access control
 - mechanisms to limit or propagate permission to access information
- expect intensive generalization of policy among organizations
- overall policy -- depends upon the underlying capabilities ascribed to the network
- based upon the identity of the entity requesting service
 - Host
 - Users

Network Supporting Policies

Supporting Policy Issues:

- additional capabilities relating to accountability of individuals for security-relevant activities
- provide environment for enforcement and monitoring of MAC and DAC policies
- two major sub-categories

Identification and Authentication

- supports MAC and DAC
- authenticates ID and clearance
- basis for determining group membership for DAC

Audit

- security relevant events are uniquely associated with a user
- hold users accountable for actions
- for network, must formulate mutually acceptable set of overall supporting policies
- likely to be harder than DAC

Network Security Policy Issues

Formal Security Policy Model

- starting point for a chain of arguments leading to higher assurance
- form of model may be influenced by technical characteristics of system to be built
 - want intuitive resemblance to subjects, objects and access characteristics of intended implementation

For each component in the network a Reference Monitor is needed.

- each Reference Monitor should have a Formal Security Policy Model (at Class B2 and above)
- do not need a Formal Security Policy Model for the entire network system
 - instead must argue that each model represents the overall security policy

Security Policy Summary

Must have a priori policy statements.

For Class B2 and above, must have Formal Security Policy Models.

Formal Security Policy Models are not required for supporting policy components.

Network TCB -- Introduction

Objectives

1. a subject is confined to a single network component for its lifetime
2. a subject may directly access only objects within its component
3. every component contains a component Reference Monitor which mediates all access which are made locally.
4. communications channels which link components do not compromise information

If a network succeeds in achieving the points described above, then its collection of Reference Monitors constitute a comprehensive network

Reference Monitor

- all network accesses are mediated
- there are no non-local accesses
- the network Reference Monitor cannot be tampered with
- no component reference monitor can be tampered with.

We need to design the network so that the above axioms can be validated.

Method to Achieve Objectives

Confine Subjects

Subjects must be confined to a single component.

notion of a <process, domain> pair for a subject

limit objects to the same component

- must assure that no domain encompasses objects from another component
- remote processes result in the creation of a new subject in the remote component.

Objects in Local Component

Subjects can directly access objects only within the component with which the subject is associated.

What about information being transmitted between components?

- information ``in motion'' is not treated as an object
- if it is not an object then it cannot be accessed until it ``comes to rest''

Components Contain Component Reference Monitor

- in some components a degenerate component reference monitor may suffice
 - this means that if the component is single level the reference monitor is degenerate
 - no accesses need be checked because
 - 1.all objects have the same label
 - 2.all subjects have the same class
- recall that each component reference monitor needs to enforce only the policy pertinent to the particular component's local accesses

Special Concerns for Distributed TCBs

Fragmented TCB Domain

Trusted Paths Between Components

Trusted Protocols

Fault tolerance

Fragmented TCB Domain

Monolithic system

- all aspects of security state are local
- all aspects of security state are immediately available
- state transitions are well defined

Distributed system

- many devices
- maintaining integrity of TCB more difficult
- A single device may not comprise totality of all system integrity constraints
 - Delays possible
 - No guarantees of stability
- Concurrent transitions at various locations may not permit total ordering of state transitions
- Security state may be replicated, consistency must be maintained
- Labels must be consistent
- Methods for comparing labels must be consistent

Trusted Paths Between Components

Must provide assurance that TCB data is passed in a trusted path

Trusted Path must provide

- message received from trusted path originated from a trusted source
- message received from trusted path was not modified
- labels of messages sent on the trusted path have not been altered
- message ordering is preserved on pair-wise trusted paths
 - prevents replays
 - this may be optional

Trusted Protocols

Protocol interpreters at different OSI levels may need to be trusted

Implementation of Trusted Path

- Insert into protocol interpreters at Transport level or below
 - delivery assurances may be part of protocol
 - expensive to verify
- Or, use cryptographic authentication with end-to-end transport level protocol
 - need not guarantee delivery
 - not too expensive to verify

Implement System-level atomic State Transitions

- May need application level protocols for making global state transitions.
- Very difficult to verify

System-level Concurrency Control

- Use to achieve atomic state transitions

Fault tolerance

Everything may not function all of the time

Must have fail-secure properties

even though something is not working, the system is still secure

just show that failure states are secure

- do not worry about the “usual” safety issues
 - progress
 - termination
 - delivery of service

Denial of Service

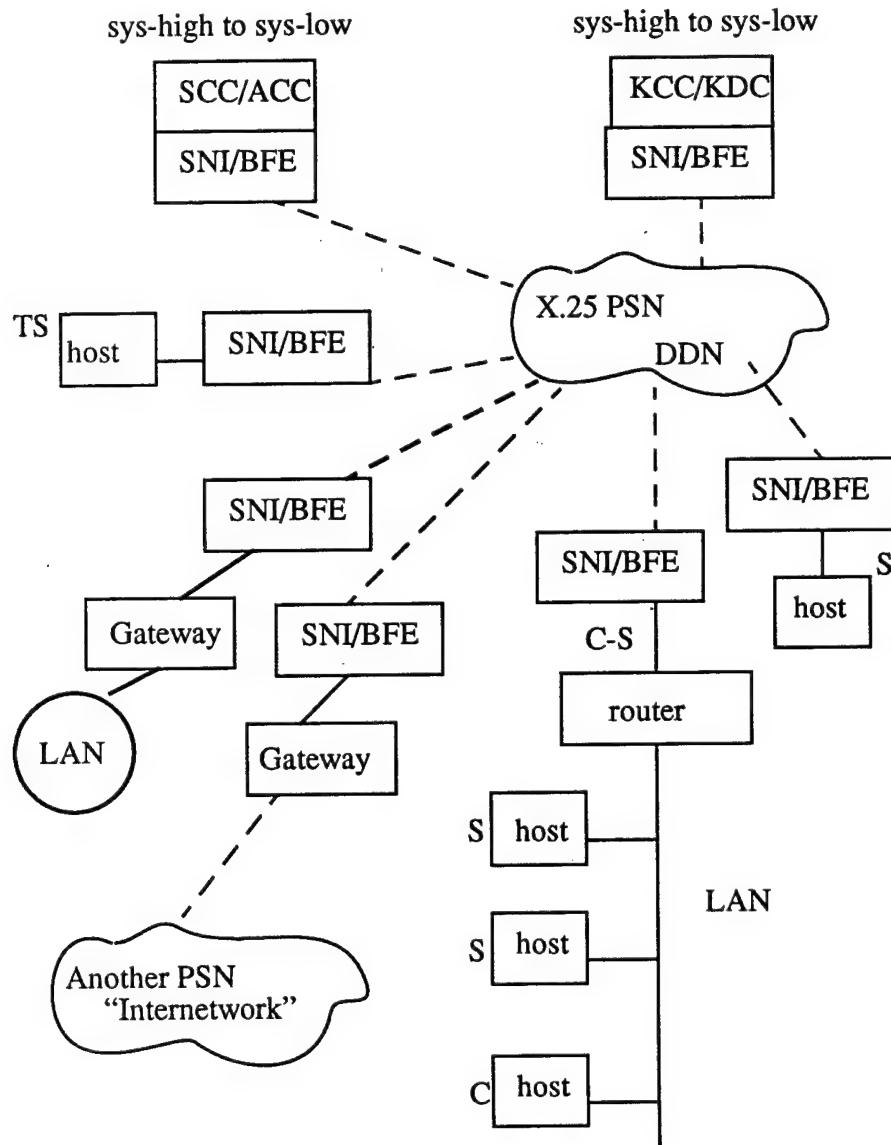
- People do worry about this
- System should be designed to provide partial policy enforcement in the face of failures
 - example: DAC could still work even though only one security level might be serviced within a LAN on a distributed system
- Traditional fault-tolerant techniques can be applied to TCB relevant data

BLACKER

Multilevel secure, with compartments

Class A1 (but pre-TNI)

Complete development but future use problematic



Blacker Overview

Secure System Applique to Defense Data Network (DDN)

Major Components

Blacker Front End

Smart encryption box between host and Packet Switch

Blacker Initialization Parameters Carrier

hand-held device for initializing BFEs

Access Control Center (ACC)

- maintains authorization tables
- controls permissions for hosts to exchange messages
- security officer activity
- maintains audit trails
- triggers auditing alarms

Key Distribution Center (KDC)

distributes encryption keys to BFEs under direction of ACC

Blacker “Domain”

- 1000 BFE's
 - These correspond to hosts
- Single
 - ACC
 - KDC
 - (Note that for reliability ACCs and KDCs were replicated)

Red/Black Separation achieved through COMPUSEC

- Security Kernel used to separate Red and Black
- Traditional has crypto device “sandwiched” between Red and Black
 - Double the hardware cost
 - Must synchronize Red/Black -- difficult
- Kernel
 - Separates Red/Black
 - Manages Crypto as MLS device

Secure Network Interface

SNI, AKA Blacker Front End (BFE)

Transparent:

Network interface presented to host

Host interface presented to network

Separates

Host-to-host shared keys

Security Levels

Enforce a connectivity property

A host may send or receive messages over a crypto connection only if it has current access to that crypto connection

Hosts Supported

Periods Processing

Multilevel

Automatic Crypto Key Support

obtains keys on demand

rekeys connections

Obtains permissions from SCC/ACC and through them enforces access control

When SCC/ACC and KCC/KDC are unavailable, emergency mode for secure communications

Connection Database

new connections added only if received by trusted path from SCC via KCC

Access Control Center

Security Control Center, a.k.a Access Control Center (ACC)

Mediates

- request for connection between source and destination
- request for mediation comes from SNI/BFE
- Message security level must be within ranges of source and destination
- Source and destination must be on each others DAC lists
- If successful will request that KCC/KDC generate keys for the connection

Security Administration

- Access Control Database
- Security Administration Interface
- Generates data to initialize SNIs
- Maintains configuration database of host/SNI sites

Within a Domain

- Replicated SCCs
- two phase commit used for consistency

Key Distribution Center

KCC, a.k.a. Key Distribution Center

All keys are encrypted

- Must communicate with BFEs at hosts' security level
 - therefore MLS
 - Use crypto seals for all imports/exports associated with KDC
 - Cryptoseals makes KDC similar to a security Guard

Centralized Key Management

Distribution of Keys controlled by Access Control Center

Separate administration of COMSEC and Access Control concerns

This page is intentionally blank.

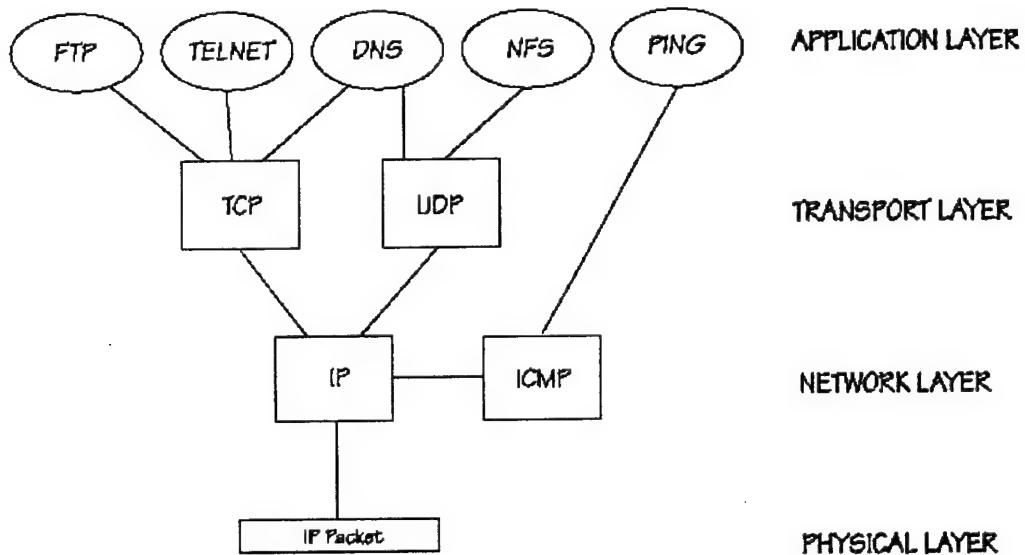
Section 10

Network Security in Today's Environment

Overview of TCP/IP Internals

The Protocols

- TCP/IP is a suite of protocols including TCP and IP, UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol), and several others.
- TCP/IP protocol suite does not conform exactly to the Open Systems Interconnection's seven layer model, but rather is pictured as shown:



Overview of TCP/IP Internals

IP

- The IP layer receives packets delivered by lower-level layers and passes the packets ``up'' to the higher-layer TCP or UDP layers, and also transmits packets that have been received from the TCP or UDP layers to the lower-level layer.
- IP packets are unreliable datagrams because IP does nothing to ensure that IP packets are delivered in sequential order or are not damaged by errors.
- The IP packets contain the source address of the host from which the packet was sent, and the destination address of the host that is to receive the packet.
- TCP and UDP services generally assume that the source address in a packet is valid when accepting a packet.
- IP address forms the basis of authentication for many services; the services trust that the packet has been sent from a valid host and that host is indeed who it says it is.
- IP contains an option known as IP Source Routing, which can be used to specify a direct route to a destination and return path back to the origination. A source routed IP packet, to some TCP and UDP services, appears to come from the last system in the route as opposed to coming from the true origination.
- IP source address is problematic and can lead to break-ins and intruder activity; furthermore it can be used to trick systems into permitting connections from systems that otherwise would not be permitted to connect.

Overview of TCP/IP Internals

TCP

- If the IP packets contain encapsulated TCP packets, the IP software will pass them "up" to the TCP software layer. TCP sequentially orders the packets and performs error correction, and implements virtual circuits, or connections between hosts. The TCP packets contain sequence numbers and acknowledgments of received packets so that packets received out of order can be reordered and damaged packets can be retransmitted.
- Connection oriented services, such as TELNET, FTP, rlogin, X Windows, and SMTP, require a high degree of reliability and therefore use TCP. DNS uses TCP in some cases (for transmitting and receiving domain name service databases), but uses UDP for transmitting information about individual hosts.

UDP

- UDP interacts with application programs at the same relative layer as TCP. However, there is no error correction or retransmission of misordered or lost packets. UDP is therefore not used for connection-oriented services that need a virtual circuit. It is used for services that are query-response oriented, such as NFS.
- It is easier to spoof UDP packets than TCP packets, since there is no initial connection setup (handshake) involved.

ICMP

- ICMP (Internet Control Message Protocol) is at the same relative layer as IP; its purpose is to transmit information needed to control IP traffic. ICMP redirect messages inform hosts about more accurate routes to other systems, whereas ICMP unreachable messages indicate problems with a route. Additionally, ICMP can cause TCP connections to terminate "gracefully" if the route becomes unavailable (*ping* is a commonly-used ICMP-based service).
- ICMP redirect messages can be used to trick routers and hosts acting as routers into using "false" routes; these false routes would aid in directing traffic to an attacker's system instead of a legitimate trusted system. This could in turn lead to an attacker gaining access to systems that normally would not permit connections to the attacker's system or network.

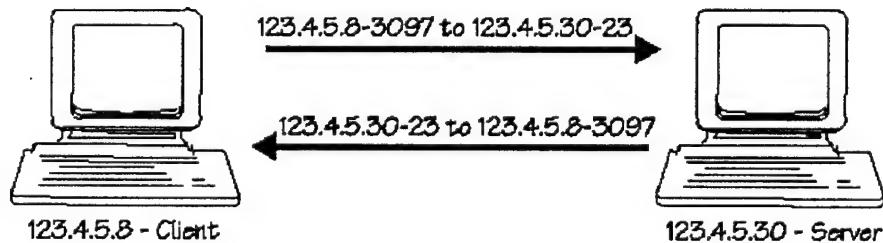
Overview of TCP/IP Internals

TCP and UDP Port Structure

- TCP and UDP services generally have a client-server relationship. (example - TELNET)
- A TCP or UDP connection is uniquely identified by
 - source IP address - the address of the system that sent the packet
 - destination IP address - the address of the system that receives the packet
 - source port - the connection's port at the source system
 - destination port - the connection's port at the destination system.
- [There is a somewhat-uniform rule that only privileged server processes, i.e., those processes that operate with UNIX superuser privileges, can use port numbers less than 1024 (referred to as privileged ports).]

Example - how ports are used for sending and receiving messages

The TELNET server listens for incoming messages on port 23, and sends outgoing messages to port 23. A TELNET client, on the same or different system, would first request an unused port number from the operating system, and then use this port when sending and receiving messages. It would place this port number, say 3097, in packets destined for the TELNET server so that the server, when responding to the client, could place the client's port number in its TCP packets. The client's host, upon receiving a message, would examine the port and know which TELNET client should receive the message.



Internet security problems

Contributing Factors:

- Internet was not designed to be very secure.
- Phenomenal success of the Internet.
- Introduction of different types of users.
- Vulnerable TCP/IP services - a number of the TCP/IP services are not secure and can be compromised by knowledgeable intruders.
- Services used in the local area networking environment for improving network management are especially vulnerable.
- Ease of spying and spoofing - the majority of Internet traffic is unencrypted; e-mail, passwords, and file transfers can be monitored and captured using readily-available software, intruders can then reuse passwords to break into systems.
- The role and importance of system management is often short-changed in job descriptions, resulting in many administrators being, at best, part-time and poorly prepared
- Lack of policy - many sites are configured unintentionally for wide-open Internet access without regard for the potential for abuse from the Internet; many sites permit more TCP/IP services than they require for their operations and do not attempt to limit access to information about their computers that could prove valuable to intruders.
- Complexity of configuration - host security access controls are often complex to configure and monitor; controls that are accidentally misconfigured often result in unauthorized access.

Internet security problems

Use of weak, static passwords.

- Passwords can be ``cracked'' a number of different ways, however the two most common methods are by cracking the encrypted form of the password and by monitoring communications channels for password packets.
- The UNIX operating system usually stores an encrypted form of passwords in a file that can be read by normal users. The password file can be obtained by simply copying it or via a number of other intruder methods. Once the file is obtained, an intruder can run readily-available password cracking programs against the passwords to obtain passwords that can be used to gain access into the system.

Host Authentication

- Some TCP or UDP services are able to authenticate only to the granularity of host addresses and not to specific users.
 - For example, an NFS (UDP) server cannot grant access to a specific user on a host, it must grant access to the entire host. The administrator of a server may trust a specific user on a host and wish to grant access to that user, but the administrator has no control over other users on that host and is thus forced to grant access to all users (or grant no access at all).

Ease of Spying/Monitoring

- When a user connects to his/her account on a remote host using TELNET or FTP, the user's password travels across in plaintext making the passwords susceptible to direct packet monitoring or network sniffers.
- Electronic mail - Most users do not encrypt e-mail, yet many assume that e-mail is secure and thus safe for transmitting sensitive information.
- The X Window System permits multiple windows to be opened at a workstation, along with display of graphics and multi-media applications (for example, the WWW browser Mosaic). Intruders can sometimes open windows on other systems and read keystrokes that could contain passwords or sensitive information.

Internet security problems

Ease of Spoofing

- Recall that the IP address of a host is presumed to be valid and is therefore trusted by TCP and UDP services. Using IP source routing, an attacker's host can masquerade as a trusted host or client.
An example:
 1. the attacker would change her host's IP address to match that of the trusted client,
 2. the attacker would then construct a source route to the server that specifies the direct path the IP packets should take to the server and should take from the server back to the attacker's host, using the trusted client as the last hop in the route to the server,
 3. the attacker sends a client request to the server using the source route,
 4. the server accepts the client request as if it came directly from the trusted client and returns a reply to the trusted client,
 5. the trusted client, using the source route, forwards the packet on to the attacker's host.
- An even simpler method for spoofing a client is to wait until the client system is turned off and then impersonate the client's system. Personal computers often use NFS to obtain access to server directories and files (NFS uses IP addresses only to authenticate clients). An attacker could, after hours, configure a personal computer with the same name and IP address as another's, and then initiate connections to the UNIX host as if it were the "real" client. (likely would be an insider attack).
- Electronic mail on the Internet is particularly easy to spoof and, without enhancements such as digital signatures, generally cannot be trusted. The exchange takes place when Internet hosts exchange mail uses a simple protocol consisting of ASCII-character commands. An intruder easily could enter these commands by hand by using TELNET to connect directly to a system's Simple Mail Transfer Protocol (SMTP) port. The receiving host trusts that the sending host is who it says it is, thus the origin of the mail can be spoofed easily by entering a sender address that is different from the true address. As a result, any user, without privileges, can falsify or spoof e-mail.

Firewalls

Why a Firewall?

- The purpose of an Internet firewall is to provide a single point of defense with controlled and audited access to services, both from within and without an organization's private network.
 1. Protection from vulnerable services.
 2. Controlled access to site systems.
 3. Concentrated Security.
 4. Enhanced privacy.
 5. Logging and statistics on network use or misuse.
 6. Policy enforcement.

Issues and Problems with Firewalls

- Restricted access to desirable services, such as TELNET, FTP, X Windows, NFS.
- If unrestricted modem access is still permitted into a site protected by a firewall, attackers could effectively jump around the firewall.
- Firewalls generally do not provide protection from insider threats.
- MBONE - Multicast IP transmissions (MBONE) for video and voice are encapsulated in other packets; firewalls generally forward the packets without examining the packet contents.
- Viruses - Firewalls do not protect against users downloading virus-infected personal computer programs from Internet archives or transferring such programs in attachments to e-mail.
- Throughput - Firewalls represent a potential bottleneck, since all connections must pass through the firewall and, in some cases, be examined by the firewall.
- All eggs in single basket - A firewall system concentrates security in one spot as opposed to distributing it among systems. A compromise of the firewall could be disastrous to other less-protected systems on the subnet.

Internet security problems

Design Decisions

- There are two fundamental philosophies that determine the overall configuration of a firewall (The distinction in these two philosophies cannot be overemphasized):

“That which is not expressly permitted is prohibited.”

- The firewall must be designed to block everything, and services must be enabled on a case-by-case basis.
- This approach is the easiest from an administrator's point of view since it provides a more “fail-safe” stance and does not demand that the administrator possess exception skills in the maintaining security of the system.

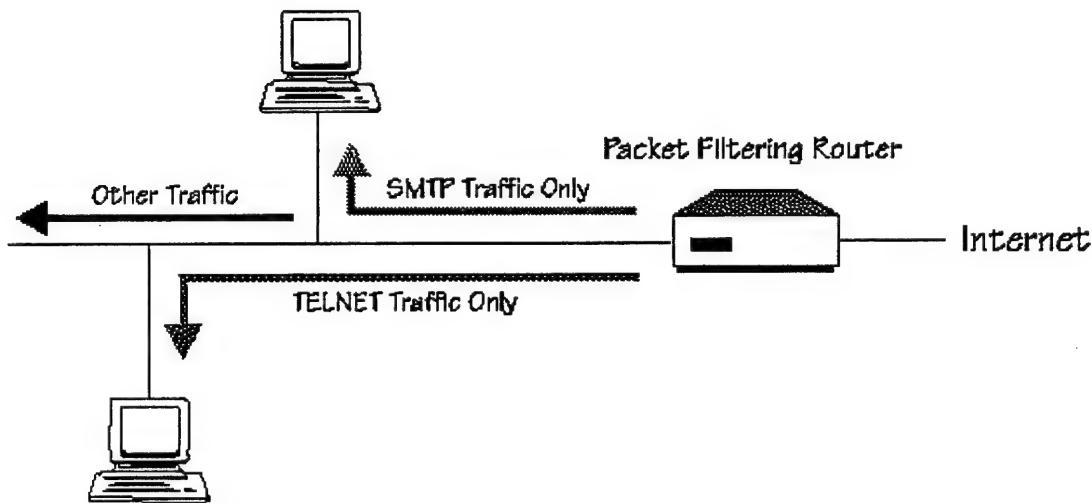
“That which is not expressly prohibited is permitted.”

- The system administrator is placed in reactive mode to the actions of the user. Although this offers the user the greatest flexibility, it often pits the user against the administrator.
- This approach requires that the administrator anticipate what users will do and requires considerable skill in auditing and maintaining the overall security of the network.
- From a security standpoint, this approach is an administrator's nightmare.

Firewall Basics

Packet Filtering

- A packet filtering router usually can filter IP packets based on some or all of the following fields:
 - source IP address,
 - destination IP address,
 - TCP/UDP source port, and
 - TCP/UDP destination port.
- A site might wish to block connections from certain addresses, or block connections from all addresses external to the site (with certain exceptions, such as with SMTP for receiving e-mail).
- If a firewall can block TCP or UDP connections to or from specific ports, then one can implement policies that call for certain types of connections to be made to specific hosts, but not other hosts.



Representation of Packet Filtering on TELNET and SMTP.

Firewall Basics

An example of packet filtering rules:

Type	Source Addr	Dest Addr	Source Port	Dest Port	Action
tcp	*	123.4.5.6	> 1023	23	permit
tcp	*	123.4.5.7	> 1023	25	permit
tcp	*	123.4.5.8	> 1023	25	permit
tcp	129.6.48.254	123.4.5.9	> 1023	119	permit
udp	*	123.4.*.*	> 1023	123	permit
*	*	*	*	*	deny

- Port 23 is the port associated with the TELNET.
- Port 25 is the port associated with the SMTP.
- Port 119 is the port associated with the NNTP.
- Port 123 is the port associated with the NTP.

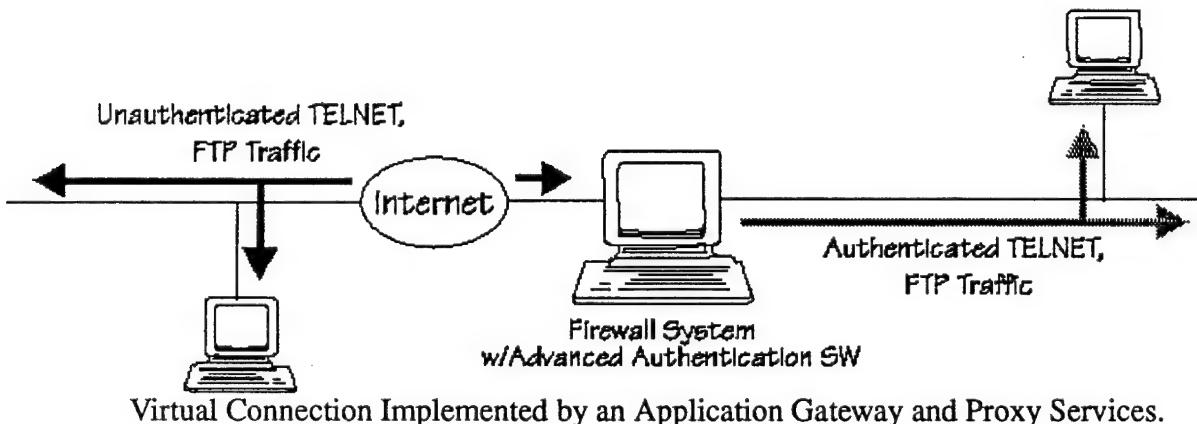
Firewall Basics

Application Gateways

- To counter some of the weaknesses associated with packet filtering routers, firewalls need to use software applications to forward and filter connections for services such as TELNET and FTP. Such an application is referred to as a proxy service, while the host running the proxy service is referred to as an application gateway.

An example of an application gateway:

- As an example, consider a site that blocks all incoming TELNET and FTP connections using a packet filtering router. The router allows TELNET and FTP packets to go to one host only, the TELNET/FTP application gateway. A user who wishes to connect inbound to a site system would have to connect first to the application gateway, and then to the destination host, as follows:
 1. a user first telnets to the application gateway and enters the name of an internal host,
 2. the gateway checks the user's source IP address and accepts or rejects it according to any access criteria in place,
 3. the user may need to authenticate herself (possibly using a one-time password device),
 4. the proxy service creates a TELNET connection between the gateway and the internal host,
 5. the proxy service then passes bytes between the two connections, and
 6. the application gateway logs the connection.



Firewall Basics

Benefits of Application Gateways

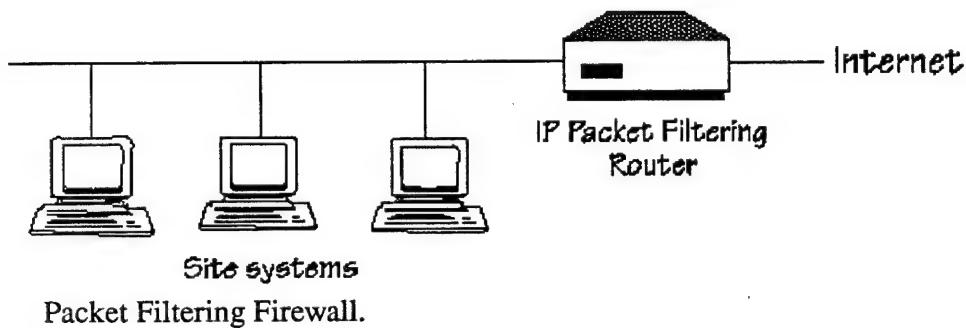
- Proxy services allow only those services through for which there is a proxy.
- The names of internal systems need not necessarily be made known via DNS to outside systems.
- Application traffic can be pre-authenticated before it reaches internal hosts and can be logged more effectively.
- Cost-effective because third-party software or hardware for authentication or logging need be located only at the application gateway.
- Rules at the packet filtering router will be less complex since the router need only allow application traffic destined for the application gateway and reject the rest.
- An e-mail application gateway serves to centralize e-mail collection and distribution to internal hosts and users. To outside users, all internal users would have e-mail addresses of the form:

user@emailhost

- The gateway would accept mail from outside users and then forward mail along to other internal systems as necessary. Users sending e-mail from internal systems could send it directly from their hosts, or in the case where internal system names are not known outside the protected subnet, the mail would be sent to the application gateway, which could then forward the mail to the destination host.
- Application gateways are used generally for TELNET, FTP and e-mail, as well as for X Windows and some other services. The application gateway can filter the FTP protocol and deny all puts to the anonymous FTP server; thus ensuring that nothing can be uploaded to the server.

Packet Filtering Firewall

- The most common and easiest to employ for small, uncomplicated sites.
- The site systems usually have direct access to the Internet while all or most access to site systems from the Internet is blocked.
- Usually, inherently-dangerous services such as NIS, NFS, and X Windows are blocked.

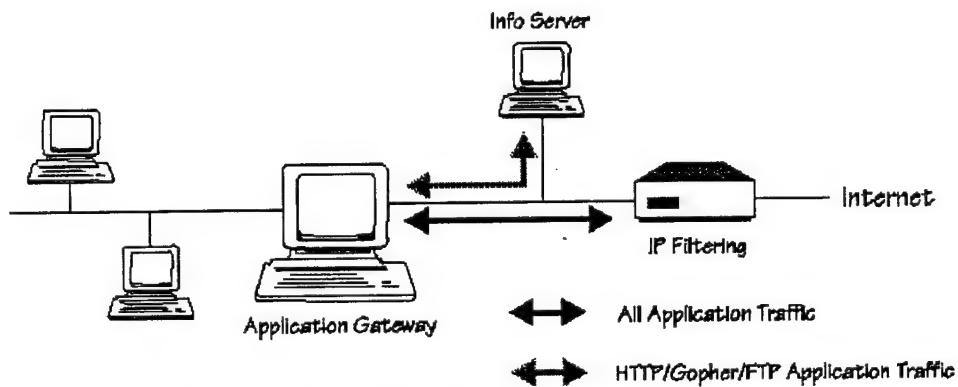


Disadvantages

- Little or no logging capability.
- Packet filtering rules are often difficult to test thoroughly.
- If complex filtering rules are required, the filtering rules may become unmanageable.
- Each host will require its own copy of advanced authentication measures.

Dual-homed Gateway Firewall

- A host system with two network interfaces with the host's IP forwarding capability disabled (i.e., the host can no longer route packets between the two connected networks).
- Services and access to site services is provided by proxy servers on the gateway.
- Simple firewall, yet very secure.
- The router can prevent direct Internet access to the firewall and force access to go through the firewall. If direct access is permitted to the server (which is the less secure alternative), then the server's name and IP address can be advertised by DNS. Locating the information server between the gateway and the router also adds to the security of the site, as any intruder penetration of the information server would still be prevented from reaching site systems by the dual-homed gateway.

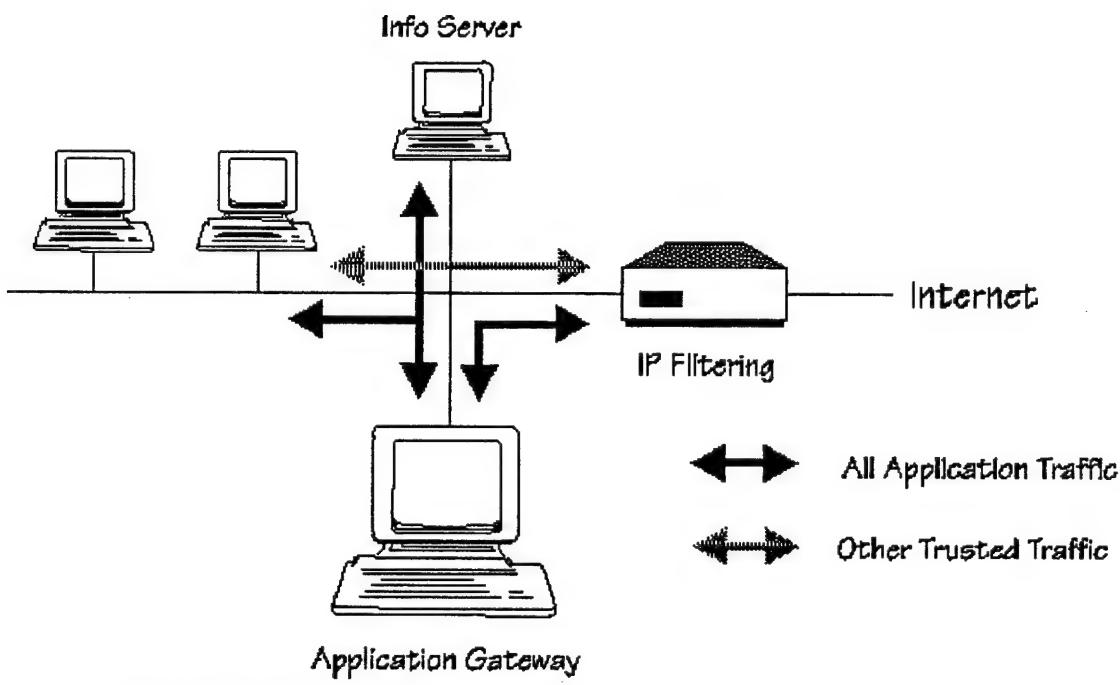


Dual-homed Gateway Firewall with Router.

- Design policy - deny all services unless they are specifically permitted, since no services pass except those for which proxies exist.
- Achieves a higher degree of privacy since the names and IP addresses of site systems are hidden from Internet systems, because the firewall does not pass DNS information.
- The firewall can house software to require users to use authentication tokens or other advanced authentication measures.
- The firewall can also log access and log attempts or probes to the system that might indicate intruder activity.
- The security of the host system used for the firewall must be very secure, as the use of any vulnerable services or techniques on the host could lead to break-ins.

Screened Host Firewall

- More flexible firewall than the dual-homed gateway firewall, but is less secure.
- Combines a packet-filtering router with an application gateway located on the protected subnet side of the router.
- The router filters inherently dangerous protocols
- Router rejects (or accepts) application traffic according to the following rules:
 - application traffic from Internet sites to the application gateway gets routed,
 - all other traffic from Internet sites gets rejected, and
 - the router rejects any application traffic originating from the inside unless it came from the application gateway.

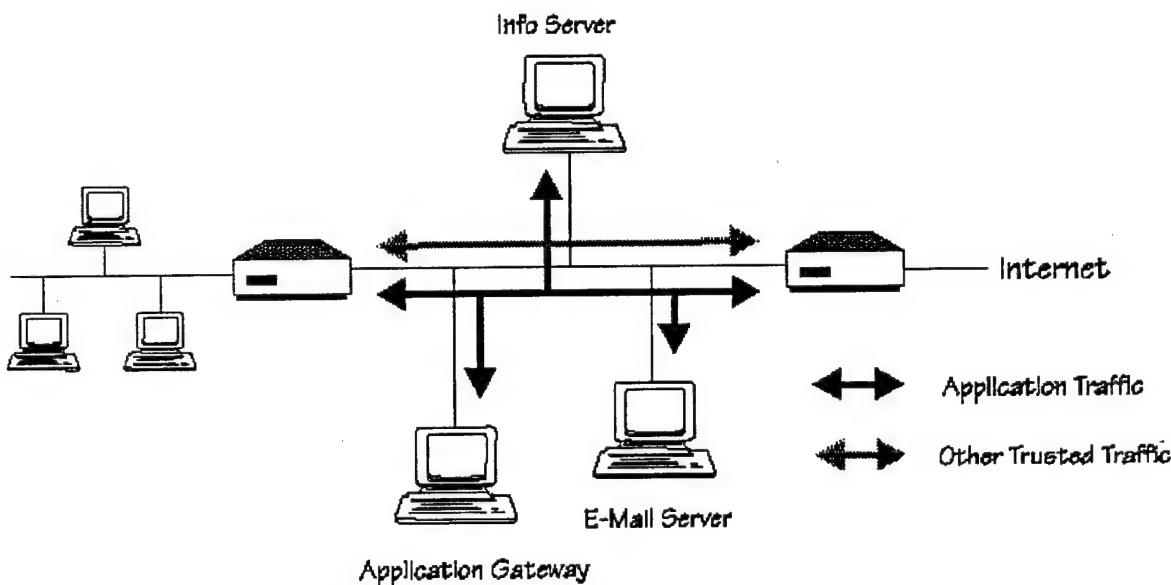


Screened Host Firewall.

- Needs only one network interface and does not require a separate subnet between the application gateway and the router.
- Permits the router to pass certain trusted services ``around'' the application gateway and directly to site systems.
- There are now two systems, the router and the application gateway, that need to be configured carefully.
- Opens up the possibility that the policy can be violated.

Screened Subnet Firewall

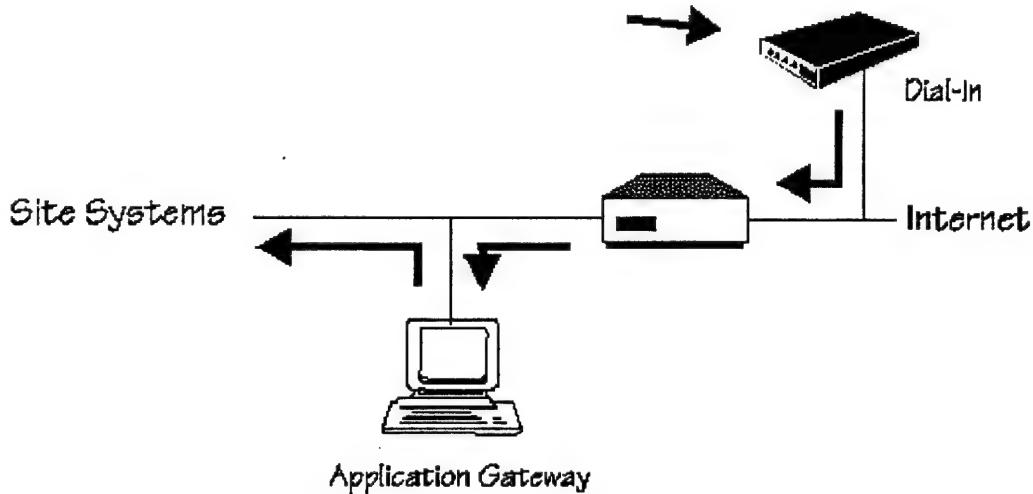
- Can be used to locate each component of the firewall on a separate system, thereby achieving greater throughput and flexibility.
- Two routers are used to create an inner, screened subnet



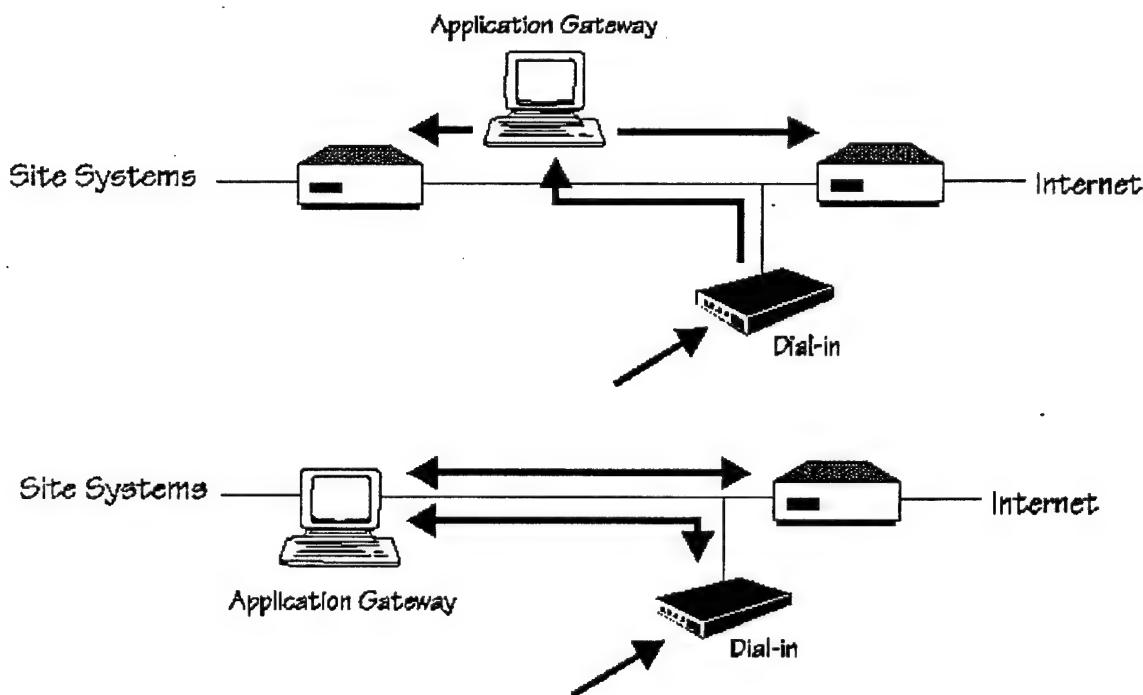
Screened Subnet Firewall with Additional Systems.

- No site system is directly reachable from the Internet and vice versa.
- Routers are used to direct traffic to specific systems, thereby eliminating the need for the application gateway to be dual-homed.
- Appropriate for sites with large amounts of traffic or sites that need very high-speed traffic.
- The two routers provide redundancy since an attacker would have to subvert both routers.

Integrating Modem Pools with Firewalls



Modem Pool Placement with Screened Host Firewall.



Modem Pool Placement with Screened Subnet and Dual-Homed Firewalls.

Section 11

Building Secure Systems II

System Evaluations

Evaluation Issues

Important Evaluation Criteria

These criteria do not specify or address how to implement the required security features.

NCSC Criteria

- Trusted Computer System Evaluation Criteria (TCSEC)
- a.k.a. Orange Book
- a.k.a. Criteria
- Ties assurance with features.
- More on following slides.

Federal Criteria for Information Technology Security

- Intended to be joint NCSC/NIST standard.
- Attempts to establish protection profiles in order to achieve greater flexibility over TCSEC.
- Rejected
 - scientifically unsound
 - politics

Common Criteria

- Currently in Draft form.
- Uses protection profiles.
- Attempt to harmonize criteria of EC, US, and Canada.
- Highly flexibility is an objective.
- Threat perspective.
- Considered by some to be dangerous because of arbitrary mix of features and assurance.

TCSEC Issues

The TCSEC defines four basic divisions; A, B, C and D.

- Class A designates the highest level of assurance of policy enforcement.
- Within a division, numbers are used to designate a finer distinction of levels, (i.e., **B1, B2, B3**).
- A greater number indicates higher assurance.
- Classes **C** through Class **B1** might be add on measures to existing operating system
 - Division **D** is failure
- At Class **B2** and above security must be included in system design.
- Class **A1** systems subjected to formal methods.
- **TCSEC class requirements are cumulative.**

The TCSEC analysis of systems is divided into four requirements areas:

- Policy
- Accountability
- Assurance
- Documentation

Each requirements area is divided into a number of finer requirements.

- Lower assurance systems (e.g., C2) must satisfy a specific set of these requirements.
- Higher assurance systems (e.g., B2) must satisfy a larger specific set of these requirements.
- The Highest assurance systems (A1) must satisfy all of these requirements.

TCSEC Requirements Chart

<i>Security Policy</i>
Discretionary Access Control
Object Reuse
Labels
Label Integrity
Exportation of Labeled Information
Labeling Human Readable Output
Mandatory Access Control
Subject Sensitivity Labels
Device Labels
<i>Accountability</i>
Identification and Authentication
Audit
Trusted Path
<i>Assurance</i>
System Architecture
System Integrity
Security Testing
Design Specification and Verification
Covert Channel Analysis
Trusted Facility Management
Configuration Management
Trusted Recovery
Trusted Distribution
<i>Documentation</i>
Security Features User's Guide
Trusted Facility Manual
Test Documentation
Design Documentation

TCSEC Requirements

Security Policy

Discretionary Access Control

Object reuse

- When a storage object (page frame, disk sector, magnetic tape, etc.) is initially assigned, allocated or reallocated to a subject, the TCB will ensure that the object contains no residual data.

Labels

- This is a requirement for labels (sensitivity or integrity) associated with each system resource (e.g., subject, object).
 - Label Integrity - Exported labels shall accurately reflect internal labels.
 - Exportation of Labeled Information - I/O devices will be labeled either single-level or multilevel.
- Labeling of Human Readable Output - The TCB will mark human readable output.

Mandatory Access Control

Subject Sensitive Labels

- The TCB will notify each terminal user of each change in the security level associated with the user.

Device Labels

- Minimum and maximum security levels will be assigned to all attached devices.

Accountability

Identification and Authentication

Audit

Trusted Path

TCSEC Requirements

Assurance

System Architecture

- The TCB shall maintain a domain for its own execution that is protected from tampering. It shall be internally structured in well-defined largely independent modules. It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not.

System Integrity

- There shall be features that can be used to periodically validate the correct operation of the hardware and firmware elements of the TCB.

Security Testing

- The security mechanisms shall be tested.

Design Specification and Verification

- Formal or informal models shall be used to verify system correctness.

Covert Channel Analysis

- The developer shall conduct a through search for covert channels and make a determination of the maximum bandwidth of each identified channel.

Trusted Facility Management

- The TCB shall support separate operator and administrator functions.

Configuration Management

- A configuration management system shall be used.

Trusted Recovery

- Procedures and/or mechanisms shall be provided that can recover a system without a comprise of protection.

Trusted Distribution

- Trusted distribution facilities shall be used.

TCSEC Requirements

Documentation

Security Features User's Guide

- A summary of protection mechanisms.

Trusted Facility Manual

- An administrator manual about running a secure facility.

Test Documentation

- Documentation of the test plan and test results.

Design Documentation

- A description of the design.

The following symbols are used in the chart on the next page.

No requirement	
New or enhanced requirement	⊗
No additional requirement	⇒

TCSEC Requirements Chart

Criteria	D	C1	C2	B1	B2	B3	A1
Security Policy							
Discretionary Access Control	⊗	⊗	⇒	⇒	⊗	⇒	
Object Reuse	⊗	⇒	⇒	⇒	⇒	⇒	
Labels		⊗	⊗	⇒	⇒	⇒	
Label Integrity		⊗	⇒	⇒	⇒	⇒	
Exportation of Labeled Information		⊗	⇒	⇒	⇒	⇒	
Labeling Human Readable Output		⊗	⇒	⇒	⇒	⇒	
Mandatory Access Control		⊗	⊗	⇒	⇒	⇒	
Subject Sensitivity Labels			⊗	⇒	⇒	⇒	
Device Labels			⊗	⇒	⇒	⇒	
Accountability							
Identification and Authentication	⊗	⊗	⊗	⇒	⇒	⇒	
Audit	⊗	⊗	⊗	⊗	⊗	⇒	
Trusted Path				⊗	⊗	⇒	
Assurance							
System Architecture	⊗	⊗	⊗	⊗	⊗	⇒	
System Integrity	⊗	⇒	⇒	⇒	⇒	⇒	
Security Testing	⊗	⊗	⊗	⊗	⊗	⊗	
Design Specification and Verification			⊗	⊗	⊗	⊗	
Covert Channel Analysis				⊗	⊗	⊗	
Trusted Facility Management				⊗	⊗	⇒	
Configuration Management				⊗	⇒	⊗	
Trusted Recovery					⊗	⇒	
Trusted Distribution						⊗	
Documentation							
Security Features User's Guide	⊗	⇒	⇒	⇒	⇒	⇒	
Trusted Facility Manual	⊗	⊗	⊗	⊗	⊗	⇒	
Test Documentation	⊗	⇒	⇒	⊗	⇒	⊗	
Design Documentation	⊗	⇒	⊗	⊗	⊗	⊗	

Class D and C1

Class D Systems: Minimal Security

- There are no evaluated systems in this class.

Class C1 Systems: Discretionary Security Protection

- C1 systems provide rather limited security features.
- C1 systems are an environment of "cooperating users processing data at the same level of security."
- Two main features:
 - I and A, e.g., passwords
 - DAC
 - Does not require a distinction between read, write and execute.
 - Allows wildcards. E.g., M* = all users whose names begin with M.

Class C2

Class C2 Systems: Controlled Access Protection

- Accountability through password controls and audit.
- More detailed discretionary controls.
- Object reuse requirement.

DON CAP Program (C2 by 92)

Controlled Access Protection (CAP) Guidebook

(NAVSO P-5239-15)

- Functional interpretation of “Class C2” requirements.
- Describes minimum set of automated controls for a system.
- All DoN systems must be assessed for CAP compliance.
- All DoN systems are considered to process “sensitive unclassified” data as a minimum and therefore must adhere to “Class C2” requirements due to data aggregation and connectivity.
- Waivers may be granted but must be reviewed annually.

CAP Assessments

Basic

- Used to determine if a set of CAP features exist in a product (documentation review)

Detailed

- Used to determine whether CAP features function as described or claimed

Recognized-Authority

- Compliance assessments by:
 - NCSC (National Computer Security Center)
 - NESSEC (Naval Electronic Sys. Security Eng. Center)
 - NRL (Naval Research Laboratory)
 - AFCSC (Air Force Cryptologic Support Center)

C2 Advantages - Abuse of authority, Direct probing.

Class B1

Class B1 Systems: Labeled Security Protection

- An informal or formal model of the Security Policy is required.
- All "major" objects are required labeled and these labels are used to enforce a MAC policy.
- B1 is often called "C2 with labels", since B1 systems do not require much more assurance than C2 systems.
- The labels must be implemented in a way such that a system has the potential to support different Human Readable Labels.
 - The internal labels are probably just numbers and a Human Readable Label Manager maps the numbers to the Human Readable Labels.
 - DoD might use Top Secret, Secret, etc.
 - Non-DoD might use Sensitive, Non-sensitive, etc.
- Requires a Security Officer.
- Requires Security Officer documentation.
 - Administration of labels
 - Securely manage user clearances.
- Requires Human Readable Labels on output.

Class B1 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Direct Probing

Some Protection Against Probing with Malicious Software

Risks

Direct Penetration

Subversion of Mechanism

Class B2

High Assurance versus Low Assurance

- Systems rated at class B1 and below are commonly referred to as "low assurance" systems.
- Systems rated at B2 and above are commonly referred to as "high assurance" systems.

Class B2 Systems: Structured Protection

- Not much additional user-visible security features.
- Instead, B2 has extended features and additional assurance that the features were designed to work properly.
- The system is relatively resistant to penetration.

Requirements:

- Formal Security Policy Model.
- MAC for all subjects and objects.
- Greater isolation for Security Kernel.
- Methodical configuration management.
 - Protects against illicit modifications.
- Greater use of modularity and use of hardware features.
- Trusted path.
- Covert channel analysis.
- Identification and isolation of non-security relevant code.
- Penetration testing will augment interface testing.

Class B2 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Direct Probing

Some Protection Against Probing with Malicious Software

Some Protection Against Direct Penetration

Risks

Direct Penetration

Subversion of Mechanism

Class B3

Class B3 Systems: Security Domains

- There are no new user-visible features.
- Must satisfy Reference Monitor implementation requirements.
 - Simple
 - Tamper-proof
 - Impossible to bypass
- Exclude code from Security Kernel that is not security relevant.
- Highly resistant to penetration.
- Trusted Facility Management.
 - Assignment of specific individual as security officer.
- Requires Trusted Recovery.

Class B3 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Protection against Direct Probing

Some Protection Against Probing with Malicious Software

Significant Protection Against Direct Penetration

Some Protection Against Subversion of Mechanism

Risks

Subversion of Mechanism

Class A1

Class A1 Systems: Verified Protection

- Pretty much functionally equivalent to B3 systems.
- Trusted Distribution is the only new feature.

Additional assurance provided by:

- Formal analysis and mathematical proof that the system design matches the system's security policy and its design specifications.
- Trusted Distribution
 - This decreases the possibility of subversion during distribution, i.e., replacement of TCB parts.
- Life-cycle configuration management.
 - Hardware
 - Software
 - Specifications
 - Development tools
- Life-cycle covers:
 - design
 - development
 - production
 - distribution
- Formal Top Level Specification can be analyzed by computer tools to find all covert storage channels.

Class A1 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Protection against Direct Probing

Protection Against Probing with Malicious Software

Increased Assurance Against Direct Penetration

Increased Assurance Against Subversion of Mechanism

TCSEC Security Requirements

Security Requirements Overview

Class	Description	Criteria
Class D	Minimal Protection	Reserved for systems that have been evaluated but fail to meet the requirements for a higher evaluation class.
Class C1	Discretionary Security Protection	Nominally satisfies the discretionary security requirements by providing separation of users and data. It incorporates some form of credible controls capable of enforcing access limitations on an individual basis, i.e., suitable for allowing users to be able to protect project or private information and to keep other users from accidentally reading or destroying their data. The class C1 environment is expected to be one of cooperating users processing data at the same level(s) of sensitivity.
Class C2	Controlled Access Protection	Enforce a more finely grained discretionary access control than C1 systems, making users individually accountable for their actions through login procedures, auditing of security-relevant events, and resource isolation.
Class B1	Labeled Security Protection	All features of class C2 are required. In addition, an informal statement of the security policy model, data labeling, and mandatory access control over named subjects and objects must be present. The capability must exist for accurately labeling exported information. Any flaws identified during testing must be removed.
Class B2	Structured Protection	Based on a clearly defined and documented formal security policy model that requires the discretionary and mandatory access control enforcement found in B1 systems to be extended to all subjects and objects in the ADP system. In addition, covert channels are addressed. The TCB must be carefully structured into protection-critical and non-protection-critical elements. The TCB interface is well defined and the TCB design and implementation enable it to be subjected to more thorough testing and more complete review. Authentication mechanisms are strengthened, trusted facility management is provided in the form of support for system administrator and operator functions, and stringent configuration management controls are imposed. The system is relatively resistant to penetration.

Security Requirements Overview

Class	Description	Criteria
Class B3	Security Domains	Must satisfy the reference monitor requirements that mediate all accesses of subjects to objects, be tamper-proof, and be small enough to be subjected to analysis and tests. To this end, the TCB is structured to exclude code not essential to security policy enforcement, with significant system engineering during TCB design and implementation directed toward minimizing its complexity. A security administrator is supported, audit mechanisms are expanded to signal security-relevant events, and system recovery procedures are required. The system is highly resistant to penetration.
Class A1	Verified Design	Functionally equivalent to B3 in that no additional architectural features or policy requirements are added. The distinguishing feature of this class is the analysis derived from formal design specification and verification techniques and the resulting high degree of assurance that the TCB is correctly implemented.

TPEP Program

NCSC Trusted Product Evaluation Program (TPEP)

- Resulted from DoD Directive 5215.1 in 1982.
- The NSA is responsible for evaluating commercial products through an independent evaluation based on TCSEC requirements by a qualified team of experts.
- TPEP phases:
 - Proposal phase
 - Vendor assistance phase
 - Design analysis phase
 - Evaluation phase
 - Rating Maintenance Phase (RAMP)

TPEP Guidelines and Interpretations

- *a.k.a. The Rainbow Series*
- Guidelines
 - A Guide to Understanding Discretionary Access Control in Trusted Systems
 - A Guide to Understanding Trusted Distribution in Trusted Systems
 - A Guide to Understanding Configuration Management in Trusted Systems
 - A Guide to Procurement of Trusted Systems
 - Guidance for Applying the DoD TCSEC in specific Environments
 - ...
- Interpretations:
 - Trusted Network Interpretation (TNI)
 - Trusted Database Interpretation (TDI)

Evaluated Products List (EPL)

- List of products that have completed evaluations and those that are in evaluation.

TPEP RAMP Program

Rating Maintenance Phase (RAMP)

- Reason for RAMP.
 - Frequency of new releases.
 - Limited evaluation resources.
- Goals
 - Keep EPL populated with trusted products.
- Approach
 - Use qualified vendor personnel (VSAs).
 - Same level of detail as original evaluation.
- Applicability of RAMP
 - All products evaluated against TCSEC, TNI, TDI.
- Scope of RAMP
 - At B2 and above NSA Future Change Review Board decides if proposed changes are appropriate for RAMP.
 - At B1 and below the Technical Review Board decides if proposed changes are appropriate for RAMP.

Balanced Assurance

Technically Sound

- Security Architectures Use TCB Subsets
- Require Greatest Assurance for Most Critical Policies
- Enforcement Mechanism for Most Critical Policies is Most Privileged

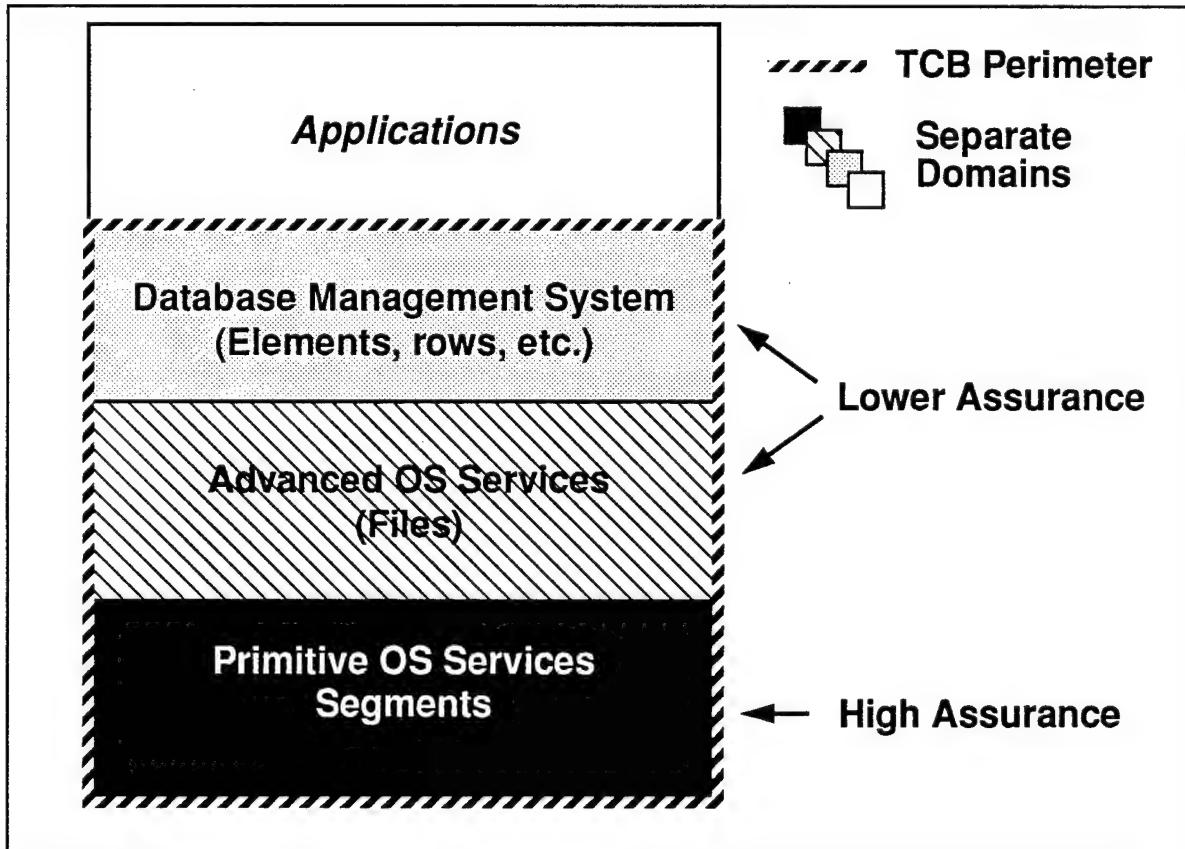
Commercially Attractive

- Can Build System Using Incrementally Evaluated Components

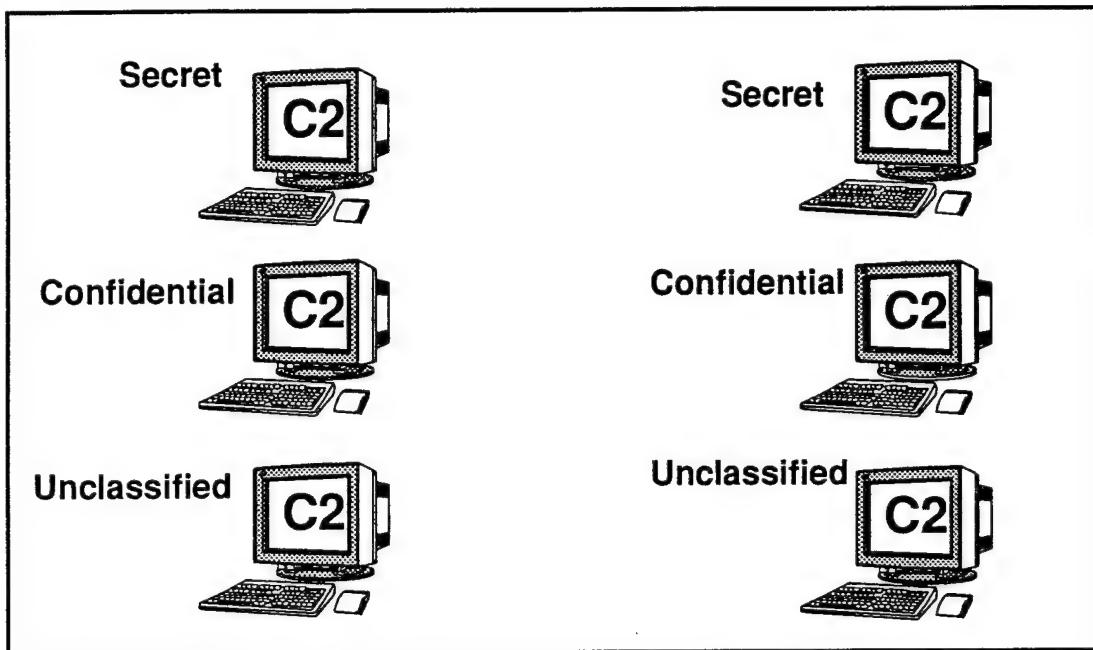
Permits High Assurance Where Critical

Does not Impose High Assurance Requirements Unnecessarily

Hierarchical Balanced Assurance Architecture

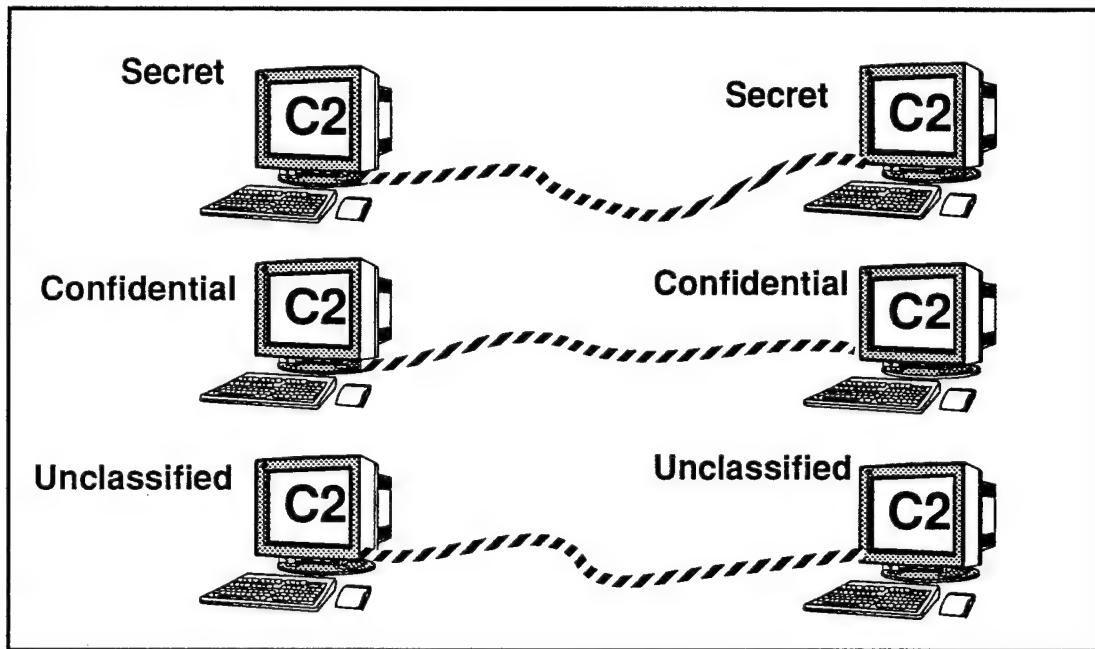


Architecture Before Networking



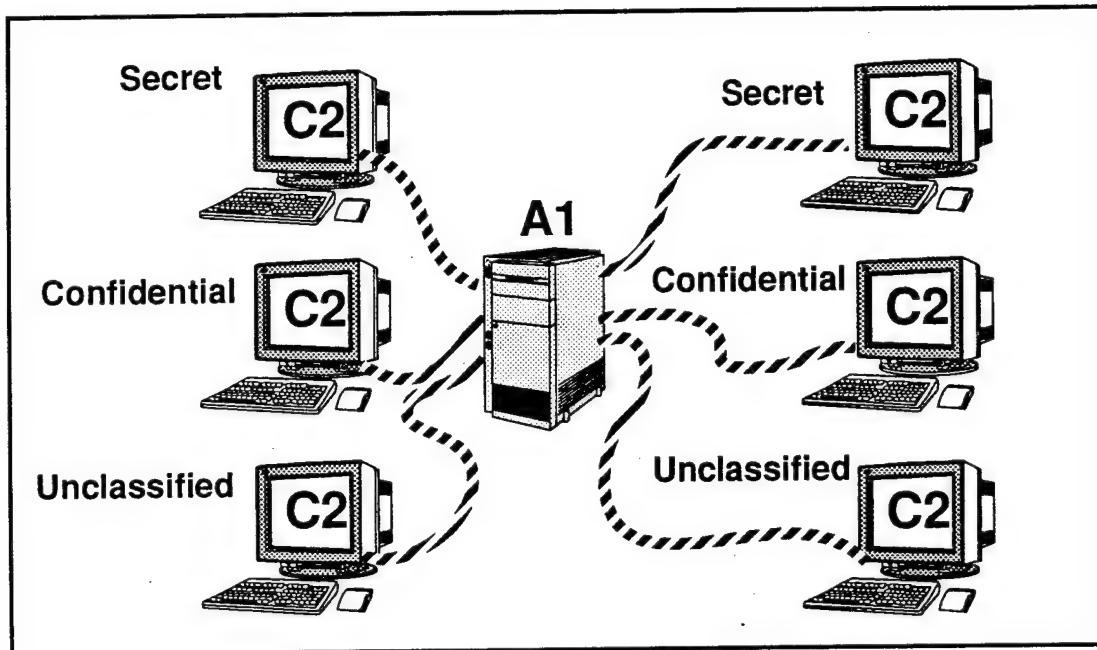
Separate systems

Single-Level Connection Architecture



Systems Connected at a Single Security Level

Networked Balanced Assurance Architecture



Systems Connected Through A High Assurance Guard

With one minor caveat, one could argue convincingly that the resultant network above could satisfy the Orange Book A1 requirements, since the mandatory enforcement is being done by an A1 rated component. The minor caveat is that A1 DAC requires the ability to deny access down to the granularity of a single user (i.e., Alice is denied read access to file X, even though file X is readable by everyone else in the world). The C2 systems shown above are not required to have this functionality.

The phrase "C2+" is often used to describe systems that satisfy C2 functionality and assurance requirements and, in addition, implement a DAC policy that has the ability to deny access down to the granularity of a single user. Thus, if C2+ systems are used in the configuration shown above, a convincing argument could be made that the entire network should warrant an A1 rating.

Basic Terms and Definitions

System Operating Modes

Dedicated Security Mode

- All users possess the proper clearance and have need-to-know for accessing all data processed and stored by the AIS.
- All information is handled at the highest level processed by the system.

System High Security Mode

- All users possess the proper security level but do not necessarily have a need-to-know.
- All information is processed at the highest level processed by the system.
- Note that the book *Computer Security Basics* (page 72) does not give the full definition.

Multilevel Security Mode

- One or more users do not possess the proper clearance for accessing the most sensitive classified data processed and stored by the AIS.
- Data labels maintained by the system can be trusted.

Controlled Security Mode

- A reduced form of multilevel security mode, when a more limited degree of trust is placed in the AIS and the classification and clearance levels are restricted.

Compartmented Security Mode

- The mode of operation which allows the system to process two or more types of compartmented information or any one type of compartmented information with other than compartmented information.

Yellow Book Provides Standard Guidance

		Maximum Data Sensitivity							
		U	N	C	S	TS	1C	MC	
Minimum Clearance or Authorization of System Users	U	C1	B1	B2	B3	*	*	*	
	N	C1	C2	B2	B2	A1	*	*	
	C	C1	C2	C2	B1	B3	A1	*	
	S	C1	C2	C2	C2	B2	B3	A1	
	TS(BI)	C1	C2	C2	C2	C2	B2	B3	
	TS(SBI)	C1	C2	C2	C2	C2	B1	B2	
	1C	C1	C2	C2	C2	C2	C2†	B1‡	
	MC	C1	C2	C2	C2	C2	C2†	C2†	

Systems for Classes C1 or C2 are assumed system high

C2† -- If users not authorized for all categories, then Class B1 or higher

B1‡ -- If 2 categories, Need Class B2

System Composition Dangers

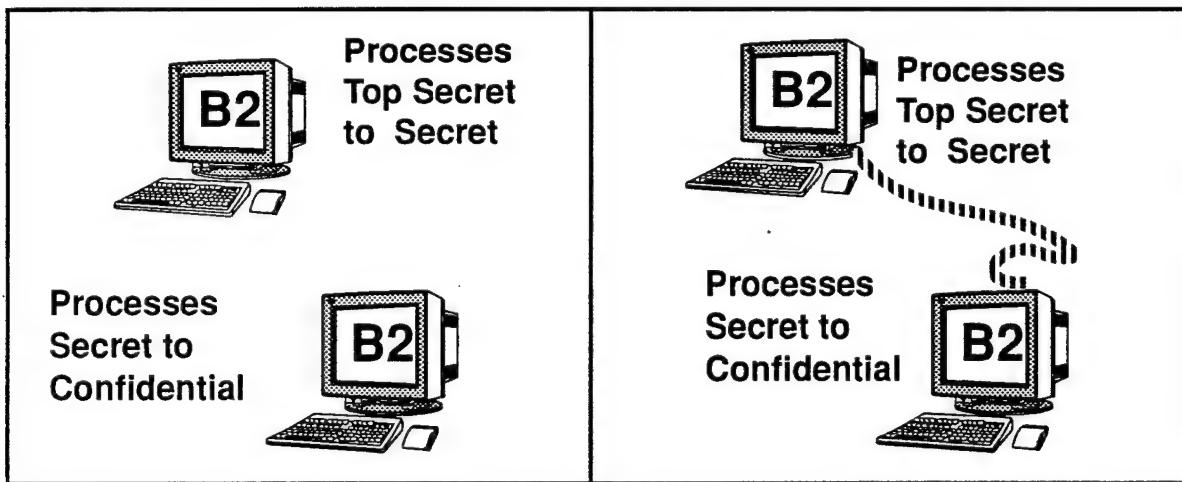
Need to Combine Systems or Components

- composition can be challenging
 - connection of separately secure systems may be insecure
 - need more research on theory of composition

Cannot Assume Adequate Assurance Although Individual Components are Sufficient for Isolated systems

Cascade Problem Example

Separate Systems have Adequate Assurance



Systems with Secret-to-Secret Connection are Inadequate

- Information can “cascade” from TS to S to C

Criteria Dangers

Mix and Match Approach

- Arbitrary Protection Profiles to Counter Specific “Threats”
- Separation of Functional Requirements from Assurance Requirements

Post Evaluation TCB Extensions

Relying on Vendor “Pedigrees”

- Most Popular University Operating System is Notably Insecure
- Example: Building Nice WYSIWYG Interfaces Does not Imply Security Competence
- Vendors May Cut Corners for Greater Profit
- Changes in Personnel or Business Strategy Unknown to Evaluation Authority
- Customer Will Not Know Assurance Lacking Until Too Late

Assurance Summary

Codify What is Demonstrated--Worked Examples

Unify with COMSEC Practice

- Utilize synergy
- Identify uses for Cryptographic Techniques -- integrity of labels
- Identify Trusted Processing -- keys

TCB Subsetting Techniques -- TNI gave a start

Trusted Subject Methods

- Covert Storage Channel Analysis
- Sufficient Design Constraints
- Definitive Tie to Cryptography

Explicitly Address Balanced Assurance

Color	Title and Summary of Contents
Orange Book	<p><i>Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TCSEC)</i></p> <p>Contains basic requirements in four categories for trusted operating systems: security policy, accountability, assurance, and documentation.</p>
Green Book	<p><i>Department of Defense (DoD) Password Management Guideline</i></p> <p>Contains a set of good practices for the design, implementation, and use of password systems used for authentication. Many trusted systems comply explicitly with this guideline.</p>
Light Yellow Book	<p><i>Computer Security Requirements - Guidance for Applying the Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TSEC) in Specific Environments</i></p> <p>Contains information on different modes of security (closed security environment, open security environment, dedicated security mode, controlled security mode, and multi-level security mode) and the “risk index” associated with each environment.</p>
Yellow Book	<p><i>Technical Rationale Behind CSC-STD-003-85: Computer Security Requirements - Guidance for Applying the Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TSEC) in Specific Environments</i></p> <p>Companion to the Light Yellow Book. Contains background information on determining the class of trusted system required for different risk indexes.</p>

Color	Title and Summary of Contents
Dark Blue Book	<p><i>Department of Defense (DoD) Magnetic Remanence Security Guideline (FOUO)</i></p> <p>Contains recommendations for using products that purge magnetic media via various types of data sanitization and magnetic remanence techniques.</p>
Tan Book	<p><i>A Guide to Understanding Audit in Trusted Systems</i></p> <p>Contains an interpretation of the auditing requirements included in the Orange Book. Auditing keeps track of sensitive activities in a system and provides a way of determining who performed these activities.</p>
Aqua Book	<p><i>Trusted Product Evaluations: A Guide for Vendors</i></p> <p>Contains procedures to follow when submitting a trusted system (or a network product, a database product, or a subsystem) to the NCSC for evaluation.</p>
Salmon Book	<p><i>A Guide to Understanding Discretionary Access Control (DAC) in Trusted Systems</i></p> <p>Contains an interpretation of the discretionary access control requirement included in the Orange Book. DAC protects files and other objects in a system at the discretion of the owner.</p>
Dark Green Book	<p><i>Glossary of Computer Security Terms</i></p> <p>Contains definitions for common terms used in government computer security publications.</p>

Color	Title and Summary of Contents
Red Book	<p><i>Trusted Network Interpretation (TNI) of the Trusted Computer System Evaluation Criteria (TCSEC)</i></p> <p>Contains an interpretation of the Orange Book requirements for networks, and a summary of specific network services: communications integrity, denial of service, and compromise protection.</p>
Coral Book	<p><i>A Guide to Understanding Configuration Management in Trusted Systems</i></p> <p>Contains an interpretation of the configuration management requirements included in the Orange Book. These requirements manage changes to the Trusted Computing Base and to the system documentation.</p>
Burgundy Book	<p><i>A Guide to Understanding Design Documentation in Trusted Systems</i></p> <p>Contains an interpretation of the design documentation requirements included in the Orange Book, including the suggested scope and level of effort for this documentation.</p>
Lavender Book	<p><i>A Guide to Understanding Trusted Distribution in Trusted Systems</i></p> <p>Contains an interpretation of the trusted distribution requirements included in the Orange Book. These requirements ensure that all elements of the TCB distributed to a customer arrive exactly as intended by the vendor. They include recommendations for packaging, security locks, courier service, etc.</p>
Venice Blue Book	<p>Contains an interpretation of the Orange Book requirements for computer security add-on products and subsystems. Subsystems typically provide features in one or more of the following categories: discretionary access control, object reuse, identification and authentication, and audit.</p>

Color	Title and Summary of Contents
Dark Red Book	<p><i>Trusted Network Interpretation Environments Guideline</i></p> <p>Companion to the Red Book. Contains information helpful when integrating, operating, and maintaining trusted computer networks, including the minimum security required in different network environments.</p>
Pink Book	<p><i>Rating Maintenance Phase (RAMP) Program Document</i></p> <p>Contains procedures for keeping an Orange Book rating up to date via the RAMP program. Participation in RAMP is required for C1, C2 and B1 systems.</p>
Purple Book	<p><i>Guidelines for Formal Verification Systems</i></p> <p>Contains procedures to follow when submitting a formal design and verification tool to the NCSC for evaluation.</p>
Brown Book	<p><i>A Guide to Understanding Trusted Facility Management</i></p> <p>Contains an interpretation of the trusted facility management requirements included in the Orange Book. These requirements mandate certain types of system and security administration - for example, the separation of operator, security administrator, and account administrator functions.</p>
Light Blue Book	<p><i>Trusted Product Evaluation Questionnaire</i></p> <p>Contains an extensive list of questions aimed at vendors of trusted systems. Examples are "What are the subjects in your system?" and "How can an operator distinguish the TCB-generated banner pages from user output?" The goal of the list is to help vendors understand what technical information is required for the system to be evaluated successfully.</p>

Color	Title and Summary of Contents
Gray Book	<p><i>Trusted UNIX Working Group (TRUSIX) Rationale for Selecting Access Control List Features for the UNIX System</i></p> <p>Contains a description of access control lists (ACLs), their use in enforcing the discretionary access control (DAC) feature included in the Orange Book, and the reasons for selecting this mechanism as a standard for trusted UNIX systems</p>
Lavender 2	<p><i>Trusted Database Management System Interpretation</i></p> <p>Contains an interpretation of the Orange Book requirements for database management systems.</p>

INITIAL DISTRIBUTION LIST

	No. Of copies
1. Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2. Dudley Knox Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5100	2
3. Research Office, Code 09 Naval Postgraduate School Monterey, CA 93943-5000	1
4. Dr. Blaine Burnham Attn: R23 National Security Agency 9800 Savage Road Fort George G. Meade, MD 20755-6000	9
5. Professor Cynthia E. Irvine Department of Computer Science Naval Postgraduate School Monterey, CA 93943	1